



AccuTerm 8 Help

Copyright 2019 Zumasys, Inc.



Table of Contents

Foreword	0
Part I Welcome to AccuTerm 8	14
1 Introduction.....	14
Features of AccuTerm 8	15
Hardware and Software Requirements	17
Installing AccuTerm 8	17
Upgrading to AccuTerm 8	18
2 Running AccuTerm.....	20
Reconnect	21
ReZume	21
3 Product Activation.....	23
4 AccuTerm User Interface.....	24
Keyboard Commands	25
5 AccuTerm Main Menu.....	26
6 Configuring AccuTerm.....	32
General Settings	32
File & Folder Settings	33
Connection Settings	34
Device Settings:Telnet.....	35
Device Settings: SSH.....	36
Key Management.....	38
Device Settings: SSL/Telnet.....	40
Device Settings: Serial Port.....	41
Device Settings: Modem.....	42
ReZume Settings.....	44
Terminal Settings	44
Screen Settings.....	46
Color Settings.....	48
Attribute Colors	50
Palette Settings	50
Background Settings.....	51
Keyboard Settings.....	51
Keyboard Programming.....	52
Font Settings.....	53
Sound Settings.....	54
Printer Settings	54
Clipboard Settings	57
Mouse Settings	58
File Transfer Settings	58
TCL Rules	59
Window Settings	60
7 Customizing the Menu and Toolbar.....	62
Menu Properties	63
Band Properties	63
Tool Properties	64
Image Selection	66

8	File Transfer.....	67
	File Download	67
	File Upload	68
	Import & Export MultiValue Data	69
9	Data Capture.....	71
10	Troubleshooting.....	72
11	Contacting Zumasys.....	73
Part II	AccuTerm 8 Automation	74
1	Macro Recorder.....	74
2	Scripting.....	75
	Running a Script	76
	Debugging a Script	77
	Controlling AccuTerm with Scripts	78
	Script Language Extensions	79
	Extensions.....	80
	AppActivate statement.....	80
	AppClose statement.....	80
	AppFind function.....	80
	AppGetActive function.....	80
	AppGetPosition statement.....	80
	AppGetState function.....	81
	AppHide statement.....	81
	AppList statement.....	81
	AppMaximize statement.....	81
	AppMinimize statement.....	81
	AppMove statement.....	81
	AppRestore statement.....	82
	AppSetState statement.....	82
	AppShow statement.....	82
	AppSize statement.....	82
	Chain statement.....	82
	Command() function.....	83
	FileExists function.....	83
	Item function	83
	ItemCount function.....	83
	Line function	83
	LineCount function.....	83
	OpenFileName function.....	84
	Pause statement.....	84
	Random function.....	84
	SaveFileName function.....	84
	Shell function	84
	Sleep statement.....	85
	Word function	85
	WordCount function.....	85
3	Object Reference.....	86
	The AccuTerm Object	87
	AccuTerm properties.....	88
	ActiveSession property.....	88
	AutoClose property.....	88

Common property.....	88
Common collection.....	89
CustomMouseTable property.....	89
FuncBarPos property.....	89
FuncBarStyle property.....	89
FuncBarVisible property.....	90
Height property.....	90
Largelcons property.....	90
LayoutFilename property.....	90
Left property	90
Menu property.....	91
NoCloseWarning property.....	91
PhoneBookName property.....	91
ProductLicenseType property.....	91
ProductName property.....	91
ProductRelease property.....	92
ProductSerialNumber property.....	92
RecentListSize property.....	92
RegisteredCompany property.....	92
RegisteredLocation property.....	92
RegisteredUser property.....	93
SessionBarVisible property.....	93
Sessions property.....	93
ShortcutNewWindow property.....	93
SingleInstance property.....	93
StatusLineVisible property.....	94
ToolbarVisible property.....	94
Top property	94
TrackKeyboardState property.....	94
Visible property.....	95
Width property.....	95
WindowState property.....	95
AccuTerm methods.....	95
Activate method.....	95
Arrange method.....	95
Close method	96
Hide method	96
Move method	96
OpenLayout method.....	96
Resize method.....	97
SaveLayout method.....	97
SettingsDialog method.....	97
Show method	98
Terminate method.....	98
The Menu Object	98
Menu properties.....	99
Count property.....	99
Item property	99
MnuBands property.....	99
The MnuBand Object.....	99
MnuBand properties.....	100
BandType property.....	100
Caption property (MnuBand).....	100
Count property (MnuBand).....	100

DockingArea property.....	100
MnuTools property.....	101
Name property.....	101
Visible property (MnuBand).....	101
The MnuTool Object.....	101
MnuTool properties.....	102
Action property.....	102
Caption property (MnuTool).....	102
Checked property.....	102
Enabled property.....	102
ToolID property.....	103
ToolTipText property.....	103
Visible property (MnuTool).....	103
The ScreenBlock Object	103
The Server Object	103
Server properties.....	104
AccountName property.....	104
ErrorMode property.....	104
LastError property.....	105
LastErrorMessage property.....	105
MDName property.....	105
ServerID property.....	105
ServerName property.....	105
UnicodeDelimiters property.....	106
UserName property.....	106
Server methods.....	106
AddItem Method.....	106
CallSub method.....	106
Connect method.....	106
DeleteItem method.....	107
Disconnect method.....	107
Download method (Server).....	107
Execute method.....	107
Export method.....	107
FileExists method.....	108
IConv method.....	108
Import method.....	108
IsConnected method.....	109
ItemExists function.....	109
OConv method.....	109
ReadItem method.....	109
Readnext method.....	109
Upload method (Server).....	110
UnlockItem method.....	110
WriteItem method.....	110
The Session Object	110
Session properties.....	115
Ansi8Bit property.....	115
AnsiAppCursor property.....	115
AnsiAppKeypad property.....	115
Answerback property.....	115
AsciiDelay property.....	116
AsciiEOL property.....	116
AutoAnswer property.....	116

AutoClose property (Session).....	116
BackgroundPictureFile property.....	116
BackgroundPictureMode property.....	117
BackgroundTransparency property.....	117
Baud property.....	117
BkspSendsDel property.....	118
BoldFont property.....	118
BreakKeyAction property.....	118
BytesIn property.....	118
BytesOut property.....	119
CapsLock property.....	119
Caption property.....	119
CaptureFileName property.....	119
CaptureMode property.....	119
CaptureSource property.....	120
Changed property.....	120
Charset property.....	120
ClientCertificateID property.....	120
Col & Row properties.....	121
Color property.....	121
Colors() property.....	121
Cols & Rows properties.....	122
Connected property.....	122
ConnectTimeout property.....	123
CopyPasteShortcut property.....	123
CopySelMode property.....	123
CursorType property.....	123
CustomMouseTable property (Session).....	124
DataBits property.....	124
DefaultCaptureDir property.....	124
DefaultXferDir property.....	124
DefaultXferMode property.....	124
DefaultXferOverwrite property.....	125
Device property.....	125
DeviceLicenseMode property.....	125
DialStatus property.....	125
DisableAppMode property.....	126
Duplex property.....	126
ExtCols & ExtRows properties.....	126
ExtMode method.....	127
FileName property.....	127
FKeys() property.....	127
FontName property.....	128
FontSize property.....	128
GmodeEnable property.....	128
Handshake property.....	128
Height property (Session).....	129
HistoryRows property.....	129
HostCertificateValidation property.....	129
HostKeyFingerprint property.....	130
HostKeyValidation property.....	130
HostName property.....	130
HostPort property.....	130
HostTermType property.....	130

hWnd property.....	131
Icon property	131
ID property	131
InputMode property.....	131
KeepaliveInterval property.....	132
KermitCRC property.....	132
KermitEOL property.....	132
KermitErrRetry property.....	132
KermitInitRetry property.....	132
KermitQuote property.....	133
KermitRept property.....	133
KermitTimeout property.....	133
Left property (Session).....	133
LegibleFont property.....	134
LockBaudRate property.....	134
LockFKeys property.....	134
LockKeyboard property.....	134
MapUpperFKeys property.....	134
Menu property (Session).....	135
MessageRows property.....	135
MouseEnable property.....	135
NoAutoWrap property.....	135
NormCols & NormRows properties.....	136
NormMode method.....	136
OnLine property.....	136
OverrideModemConfig property.....	136
Page property.....	136
Pages property.....	137
Palette() property.....	137
Parity property.....	138
PasteAllowCurlyQuotes property.....	138
PasteEOFChar property.....	138
PasteEOFMode property.....	138
PasteEOLChar property.....	139
PasteEOLMode property.....	139
PasteSuppressFinalEOL property.....	139
Port property.....	139
PrinterColorMode property.....	140
PrinterFontBold property.....	140
PrinterFontItalic property.....	140
PrinterFontName property.....	140
PrinterFontSize property.....	140
PrinterMode property.....	141
PrinterName property.....	141
PrinterOrientation property.....	141
PrinterPaperSize property.....	141
PrinterPaperSource property.....	142
PrinterTimeout property.....	143
PrintJobEject property.....	143
PrintScreenBackground property.....	143
PrintScreenEject property.....	143
ProtectAttr property.....	144
Settings method.....	144
ScaleFont property.....	144

ScreenPrintAdapterScript property.....	144
ScreenPrinterName property.....	145
ScreenPrintMode property.....	145
ScreenPrintOrientation property.....	145
ScreenPrintPaperSize property.....	145
ScreenPrintPaperSource property.....	146
ScrMode property.....	146
ScrollMode property.....	146
Selection property.....	146
ShowErrs property.....	147
SlavePrintAdapterScript property.....	147
SlavePrinterName property.....	147
SlavePrintMode property.....	147
Sound property.....	148
SSHAuth property.....	148
SSHCipher property.....	148
SSHKey property.....	148
SSHUseOnlyPreferredCipher property.....	149
SSHVersion property.....	149
StopBits property.....	149
Strip8th property.....	149
Tcpldn property.....	149
TcpNoDelay property.....	150
TelnetBinary property.....	150
TelnetBypass property.....	150
TelnetKeepalivesNOP property.....	150
TermType property.....	151
TextFileEncoding property.....	151
ThemeStyle property.....	152
Top property (Session).....	152
Visible property (Session).....	152
Width property (Session).....	152
WindowState property (Session).....	152
XferBytes property.....	153
XferFiles property.....	153
XferStatus property.....	153
XmodemTimeout property.....	153
YmodemTimeout property.....	154
ZmodemTimeout property.....	154
Session methods	154
Activate method (Session).....	154
Break method.....	154
Capture method.....	154
CaptureEnd method.....	155
Clear method.....	155
ClearSelection method.....	155
Close method (Session).....	155
Copy method.....	155
CopyHistory method.....	156
Delete method.....	156
Deselect method.....	156
Dial method	156
Download method.....	156
Emulate method.....	157

GetBlock method.....	157
GetSelection method.....	157
GetText method.....	157
Hangup method.....	158
Hide method (Session).....	158
Input method.....	158
LoadImage method.....	158
Locate method.....	159
MouseTableAdd method.....	159
MouseTableLoad method.....	159
MouseTableReset method.....	159
Move method (Session).....	159
Output method.....	160
Paste method.....	160
PlayMidi method.....	160
PlayWave method.....	160
PrinterClose method.....	160
PrinterOff method.....	161
PrinterOn method.....	161
PrintScreen method.....	161
ReadText method.....	161
Reset method.....	161
ResetComm method.....	162
ResetTerm method.....	162
Resize method (Session).....	162
Save method.....	162
SaveAs method.....	162
ScrollHistory method.....	163
Select method.....	163
SetBlock method.....	163
SetExtension method.....	163
SetSelection method.....	164
SetText method.....	164
Show method (Session).....	164
Terminate method (Session).....	164
UnloadImage method.....	165
Upload method.....	165
WaitFor method.....	165
WriteText method.....	165
Session events.....	166
DataReady event.....	166
Extension event.....	166
The Sessions Collection	166
Sessions methods.....	166
Add method	166
AddClone method.....	167
Count method.....	167
The Settings Object	167
Settings methods.....	170
Apply method.....	170
Dialog method.....	170
Initialize method.....	171
Settings_toJSON.....	171
Settings_fromJSON.....	171

Settings properties	171
Ansi8Bit property.....	171
AnsiAppCursor property.....	171
AnsiAppKeypad property.....	172
AnsiAutoprint property.....	172
Answerback property.....	172
AsciiDelay property.....	172
AsciiEOL property.....	173
AttributeMask property.....	173
AutoAnswer property.....	173
BackgroundPictureFile property.....	173
BackgroundPictureMode property.....	174
BackgroundTransparency property.....	174
Baud property.....	174
BkspSendsDel property.....	174
BoldFont property.....	175
BreakKeyAction property.....	175
Changed property (Settings).....	175
CharacterMapping property.....	175
Charset property.....	176
ClientCertificateID property.....	176
Colors() property.....	176
ConnectTimeout property.....	177
CopySelMode property.....	177
CursorType property.....	178
DataBits property.....	178
DefaultCaptureDir property.....	178
DefaultTextEncoding property.....	178
DefaultXferDir property.....	178
DefaultXferMode property.....	179
DefaultXferOverwrite property.....	179
Device property.....	179
DeviceLicenseMode property.....	179
DisableAppMode property.....	180
Duplex property.....	180
ExtCols, ExtRows properties.....	180
FKeys() property (settings).....	180
FontName property.....	181
FontSize property.....	181
GmodeEnable property.....	181
Handshake property.....	182
HistoryRows property.....	182
HostCertificateValidation property.....	182
HostKeyFingerprint property.....	182
HostKeyValidation property.....	183
HostName property.....	183
HostPort property.....	183
HostTermType property.....	183
KeepaliveInterval property.....	183
KermitCRC property.....	184
KermitEOL property.....	184
KermitErrRetry property.....	184
KermitInitRetry property.....	184
KermitQuote property.....	185

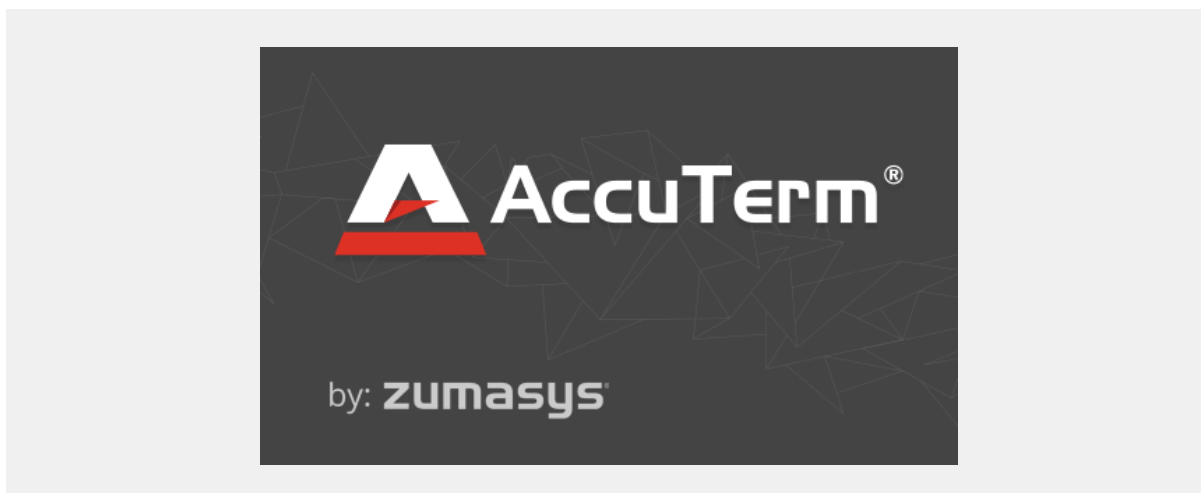
KermitRept property.....	185
KermitTimeout property.....	185
LegibleFont property.....	185
LockBaudRate property.....	185
LockFKeys property.....	186
MapUpperFKeys property.....	186
MessageRows property.....	186
MouseEnable property.....	186
NoAutoWrap property.....	187
NormCols, NormRows properties.....	187
OverrideModemConfig property.....	187
Pages property.....	187
Palette() property.....	187
Parity property.....	188
PasteAllowCurlyQuotes property.....	189
PasteEOFChar property.....	189
PasteEOFMode property.....	189
PasteEOLChar property.....	189
PasteEOLMode property.....	189
PasteSuppressFinalEOL property.....	190
Port property.....	190
PrinterColorMode property.....	190
PrinterFontBold property.....	190
PrinterFontItalic property.....	191
PrinterFontName property.....	191
PrinterFontSize property.....	191
PrinterOrientation property.....	191
PrinterPaperSize property.....	192
PrinterPaperSource property.....	192
PrinterTimeout property.....	193
PrintJobEject property.....	193
PrintScreenBackground property.....	193
PrintScreenEject property.....	194
ProtectAttr property.....	194
ScaleFont property.....	194
ScreenPrintAdapterScript property.....	194
ScreenPrinterName property.....	194
ScreenPrintMode property.....	195
ScreenPrintOrientation property.....	195
ScreenPrintPaperSize property.....	195
ScreenPrintPaperSource property.....	195
ScrMode property.....	196
ScrollMode property.....	196
SlavePrintAdapterScript property.....	196
SlavePrinterName property.....	196
SlavePrintMode property.....	197
Sound property.....	197
SSHAAuth property.....	197
SSHCipher property.....	197
SSHKey property.....	198
SSHUseOnlyPreferredCipher property.....	198
SSHVersion property.....	198
StopBits property.....	198
Strip8th property.....	199

	TcpIcn property.....	199
	TcpNoDelay property.....	199
	TelnetBinary property.....	199
	TelnetBypass property.....	199
	TelnetKeepalivesNOP property.....	200
	TermType property.....	200
	ThemeStyle property.....	201
	XmodemTimeout property.....	201
	YmodemTimeout property.....	201
	ZmodemTimeout property.....	201
	AccuTerm constants	201
4	Mouse Support.....	204
Part III	AccuTerm 8 Reference	207
1	AccuTerm Programming.....	207
2	Wyse Programming.....	217
	Wyse Tables	230
	Wyse Operating Mode Table.....	230
	Wyse Cursor Address Table.....	231
	Wyse Attribute Code Table.....	232
	Wyse Function Key Table.....	232
	Wyse Key Code Table.....	233
	Wyse Graphic Character Table.....	235
	Wyse Programming Notes	235
3	ADDS Programming.....	236
	Viewpoint Tables	239
	Viewpoint Cursor Address Table.....	239
	Viewpoint Attribute Code Table.....	240
	Viewpoint Function Key Table.....	241
	Viewpoint Key Code Table.....	241
	Viewpoint 60 Graphic Character Table.....	243
	ADDS Programming Notes	244
4	ANSI Programming.....	245
	ANSI Tables	264
	ANSI Attribute Code Table.....	264
	ANSI Function Key Table.....	265
	ANSI Extended Key Table.....	265
	ANSI Key Code Table.....	266
	ANSI Programming Notes	267
5	Pick PC Console Programming.....	268
	PC Console Color Code Table	270
6	ASCII Control Codes.....	272
7	Customizing the Installation Process.....	274
Part IV	Features for MultiValue Users	278
1	MultiValue File Transfer.....	278
	Installing the MultiValue Host Programs	279
	Alternative Host Program Installation.....	280
	Updating the MultiValue Host Programs	281
	Activating Other Accounts to use the AccuTerm Host Programs	282

Downloading Documents to your PC	282
Uploading Documents from your PC to the Host	283
FT Options and Settings	284
Downloading Data Files to your PC (export)	285
Uploading Data Files from your PC to the Host (import)	286
FTD Options and Settings	288
FTD File Types	289
Transferring Data Between MultiValue Hosts	290
Direct Host-to-Host file transfer.....	290
Host-to-Host file transfer.....	291
FTPICK Options and Settings.....	292
Integrating File Transfer Functions into Pick Applications	293
FTSEND.....	294
FTRECV.....	295
FTEXPORT.....	295
FTEXPORTDA.....	297
FTIMPORT.....	298
DOSSVC.....	299
FTS	300
2 phiReport	301
Installing phiReport	301
Activating phiReport in other accounts	302
3 MultiValue Server	303
Configuring FTSERVER	303
Using the MultiValue Server	303
4 Object Bridge	306
Initializing the Object Bridge Environment	306
Resetting the Object Bridge Environment	307
Creating an Object	307
Releasing an Object	307
Setting Object Properties	307
Getting Object Properties	307
Invoking Methods	308
Enabling Event Processing	308
Polling for Events	308
Completing Event Processing	309
Part V Copyright & Trademarks	310
Index	312

1 Welcome to AccuTerm 8

1.1 Introduction



Welcome to AccuTerm 8!

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.

Please refer to one of the following topics for detailed information about features, requirements, installation and use of AccuTerm 8:

[Features of AccuTerm 8](#)

[Hardware and Software Requirements](#)

[Installing AccuTerm 8](#)

[Upgrading to AccuTerm 8](#)

[Running AccuTerm](#)

[AccuTerm User Interface](#)

[AccuTerm Main Menu](#)

[Configuring AccuTerm](#)

[File Transfer](#)

[Data Capture](#)

[Customizing the Menu and Toolbar](#)

[Macros](#)

[Scripting](#)

[Mouse Support](#)

[Reference](#)

[AccuTerm Object Reference](#)

AccuTerm Programming

Wyse Programming

ADD5 Programming

ANSI Programming

Pick PC Console Programming
ASCII Codes
Customizing the Installation Process



AccuTerm 8 includes many features designed specifically for the MultiValue user (Pick, jBASE, QM, D3, UniVerse, UniData, Reality, etc.) We hope you find these special features useful!

[Import & Export MultiValue Data](#)

[MultiValue Server](#)

[Object Bridge](#)

GUI Development Environment

wED Editor

[Copyright & trademark notices](#)

1.1.1 Features of AccuTerm 8

Multiple sessions in multiple main windows with a new tabbed interface that supports reordering tabs, dragging tabs to their own windows and dragging tabs from one AccuTerm window to another. This allows you to run two or more main windows each with multiple tabs, all within the same process, thus using less system resources and memory. It works the same as many other programs that allow you to work with more than one document at a time like web browsers, word processors, spreadsheets, etc. so it's easy to manage. You don't have to learn a new way of working.

Simplified startup that launches a small startup panel with options to open a recent connection, browse for a connection file or open a dialog to create a new connection.

User interface has a modernized interface, 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. Toolbar, menu, session tabs and status bar can be customized by the user.

Programmer's mode: a new setting for screen size will adjust the text columns and rows to fit the window size. This feature only works with applications that respond to the "screen resize" signal, like vi.

Extended screen support of up to 240 columns by 240 rows.

Full OLE Automation / ActiveX support allows integration with other Windows applications. AccuTerm 8 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 applications where accurate terminal emulation is required.

Secure Shell connections ensure privacy using public key encryption technology.

SSL/Telnet connections ensure privacy using industry-standard SSL encryption for hosts that do not support Secure Shell.

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.

Fast file transfer capabilities using industry standard protocols like Xmodem, Ymodem, Zmodem, and Kermit. MultiValue File Transfer Wizard and utilities import and export MultiValue data in a variety of formats. And direct Host-to-Host file transfer, too!

Scripting language and macro recorder allows users to record macros or write scripts using Visual Basic for Applications (VBA).

Enhanced scroll back window saves up to 10,000 lines for instant recall.

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 Telnet, Secure Shell and SSL/Telnet supports IPV6 and Internationalized Domain Names.

GUI Development Environment helps you build user-friendly GUI front-ends for your MultiValue applications.

Image and sound support lets you to display images anywhere on the terminal screen. And you can use an image as your screen background (wallpaper). You can play Wave, MP3 and MIDI sound files.

wED Windows based program / data editor includes BASIC syntax highlighting in an MDI environment. Includes host file & item browser with an Explorer-like interface.

Tektronix graphics terminal emulation in both color and monochrome.

Visual Styles and full color mapping lets you choose any foreground and background color combination for any terminal visual attribute, and attach a border effect to the attribute. Application of Visual Styles can give character-based applications an instant face-lift.

Unicode / UTF-8 supported in all emulations, the GUI Development Environment, the wED Editor and Kermit and Zmodem file transfers.

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.

1.1.2 Hardware and Software Requirements

Memory:	1GB
Processor:	X86 or X64
Windows:	Windows 7, Windows 8, Windows 8.1, Windows 10 (32 or 64 bit) Windows Server 2012, Windows Server 2016, Windows Server 2019
Connectivity:	Network adaptor or WiFi (optional serial port or modem) Internet connection for product activation
Mouse:	Mouse or touchscreen

1.1.3 Installing AccuTerm 8

AccuTerm is available as a downloadable install file. Double-click the installation file in the 'Atwin80' folder.

To install from a downloaded install file, either choose "Run" from the "Download Complete" box, or use Windows Explorer to browse to folder where the install file was downloaded and double-click the installation file to start the installation. The installer will then guide you through the process, as outlined below in the following step-by-step instructions.

1. The **Welcome** screen is displayed. Click **Next** to begin the installation.
2. The **Installation Type** screen is displayed. Select the type of installation that you want then click the **Next** button.

Normal: installs AccuTerm 8 for all users of the computer. Program files are installed in the C:\Program Files\Atwin80 folder, and the registry is updated to establish file associations as well as component file registration. Administrator privileges are required for this type of installation. *Note: when installing on a server running Terminal Services or Citrix, you must use Control Panel "Install Application on Terminal Server" to install AccuTerm 8.*

Personal: installs AccuTerm 8 for the current user only. Program files are installed in the Atwin80 program folder and the current user's registry is updated to establish file associations as well as component file registration. If the Atwin80 folder is not in a protected directory (such as Program Files), Administrator privileges are not required for a personal installation.

Portable: installs AccuTerm 8 on a portable device such as a USB flash drive or removable hard drive. No file associations or component registrations are stored in the registry. The Portable installation allows you to take a copy of AccuTerm with you when you work on multiple machines and run it without further installation on any Windows machine running Windows 7 or later. Some of the **Setup Wizard** screens described below are not used for Portable installs.

3. The **Setup Wizard** screen is displayed. Click the **Next** button to continue.
4. The **Software License** screen is displayed. Review the license agreement, and if you agree

to the terms, click the **accept** check box and click the **Next** button.

5. The **Destination Folder** screen is displayed. Choose the directory where you want to install AccuTerm 8. The default destination depends on the **Installation Type** you selected at step 2.

Two optional items can also be selected on this screen: if you would like to create a desktop shortcut for AccuTerm 8, click the **Create shortcut** check box. If you need to enter a *custom feature code*, click the **Enable custom features** check box (most users do not require this). Click the **Next** button to continue.

6. If you selected the **Enable custom features** check box on the previous screen, the **Custom Code** screen is displayed. Enter any custom feature codes in the box on this screen, then click the **Next** button.
7. The **Ready to Install** screen is displayed. Click the **Install** button to proceed with the installation. You may need to provide Administrator credentials after clicking the **Install** button (Normal install).
8. The installation progress will be displayed during the install process. When the program has been installed click the **Finish** button to complete the installation and exit the Setup Wizard.

After installing AccuTerm 8, you can launch it from the **Start Menu** or desktop icon. When you open AccuTerm, use the **Quick Start** feature to create a new session and log in to your host system! If you selected a portable installation and opted to create a launcher, you can launch AccuTerm by double-clicking the "Start AccuTerm 8" batch file in the root of the portable drive.

If you are using a MultiValue host and you would like to take advantage of the special MultiValue features built into AccuTerm 8, you should install the [AccuTerm MultiValue Host Programs](#). If you are unsure about installing the host programs, check with your system administrator.

1.1.4 Upgrading to AccuTerm 8

When upgrading from prior versions of AccuTerm to AccuTerm 8, there are several new features of which you should be aware and some things you should know before you start using it:

- AccuTerm 8 can coexist with AccuTerm 7

When you run AccuTerm 8 for the first time, you will be given the option to import settings and preferences from a previous version of AccuTerm that is detected.

- Simplified startup: The "Welcome" screen and the "Connection Wizard" have been replaced by a small startup panel with options to launch a recently used session, open a saved session, or to configure a new connection. There is a "More settings..." button if you need access to the full setting set.
- Session files: AccuTerm 8 session profile files (with an extension of .atsp) are different from the AccuTerm 7 configuration files (with an extension of .atcf). While the AccuTerm 7 files will work with AccuTerm 8, the new session files are formatted as JSON and designed to work more efficiently with the new features and for sharing profiles between desktop, web and mobile versions.

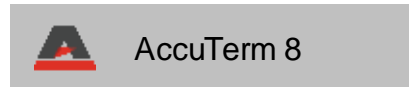
When you open an AccuTerm 7 configuration file, it is strongly recommended that you save it as the new type. When you click **File ► Save As**, AccuTerm 8 will automatically create the new session file for you. We recommend that you store the new session files in a different location to avoid re-opening old session files. Once you are comfortable with the new session files, you can remove or archive the old configuration files.

AccuTerm 8 session files will not work with AccuTerm 7.

- **Licensing:** A new licensing model is now used. There is a Zumasys license server that monitors the licensing of AccuTerm. To activate AccuTerm 8, you need to select "Activate" from the evaluation notice screen, or select the **Activate** option on the **Help** menu. You will need a valid email address and license key.
- **Tear-off tabs:** AccuTerm 8 allows you to drag tabs on the session tab bar to re-order the tabs. You can also drag tabs outside of the AccuTerm window to open them in a separate window or drag tabs between different AccuTerm windows.
- **Programmer's mode:** A new [Screen Setting](#) option for screen size, **Variable**, will adjust the screen size (text columns and rows) to fit the window size. Previously, only Normal (80 column) and Extended (132 column) were supported. This feature will send a "screen resize" signal to the server when the window is resized. Applications ((vi, JED, jsh) which respond to the "screen resize" signal should automatically adjust the screen columns and rows for the new size.
- **Toolbar icons:** All of the icons used by the main AccuTerm application have been replaced with simple monochrome icons.

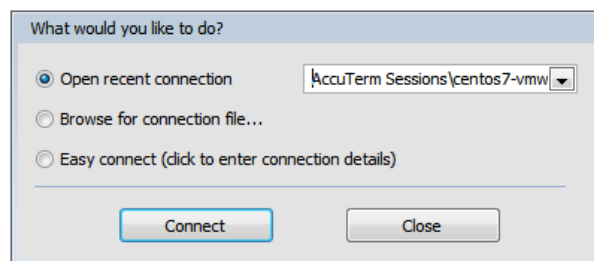
1.2 Running AccuTerm

To run AccuTerm 8, simply select AccuTerm 8 from the Start menu, or click the AccuTerm icon on the desktop. AccuTerm's main window will be displayed.



If you have a prior version of AccuTerm installed and this is the first time you have run AccuTerm 8, you will be given the opportunity to import all of your settings and preferences from the prior version into AccuTerm 8. This is a one-time offer.


Upon startup, AccuTerm 8 displays a **Quick Start** panel:

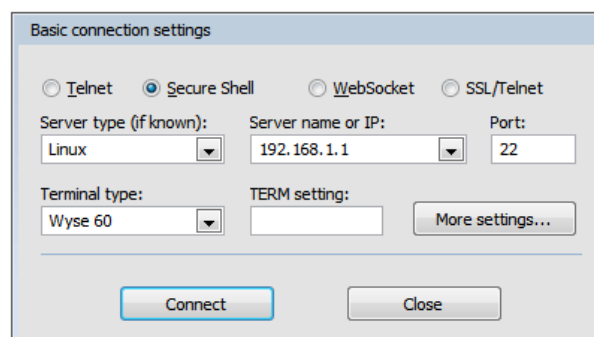


From this panel you can create a new session, open an existing session or open a recently used session. *Note: if you start AccuTerm by double-clicking a session file (.atsp) or a shortcut to a session file, the **Quick Start** panel is not displayed and the session file is opened automatically.*

Creating a New Session

Create a new session by selecting the **Easy connect** option from the **Welcome** panel, or by using **File**

► **New** from the main menu or by clicking the  button on the toolbar. The **Easy connect** option will open a **Basic connection settings** panel where you can select common connection settings to connect to your server quickly.




If you need to adjust settings which are not shown on the **Basic connection** panel, click the **More settings...** button to open the full [Settings Dialog](#).

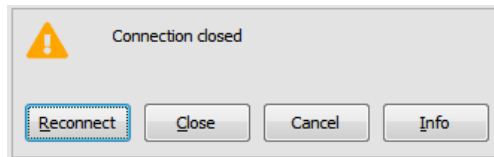
Shortcuts

After saving a session file (.atsp) or layout file (.atlp) you can create a shortcut to the file so that AccuTerm automatically opens that session or layout. To create a shortcut, use Windows Explorer to find the session or layout file (usually in the My Documents folder). Right-click on the desired file. Select **Copy** from the popup menu. Position the mouse where you want to create the shortcut (desktop, Start menu, etc.) Right-click and select **Paste Shortcut** from the popup menu. If you created the shortcut on the desktop, double-clicking the new shortcut icon will start AccuTerm and open the desired session or layout. If you created the shortcut in the Start menu, selecting the new menu item in the Start menu will do the same.

1.2.1 Reconnect

When you are connected to a server over a network (SSH, Telnet, SSL/Telnet), and the connection is terminated by the server or application (for example logging out of Linux will terminate the connection), or terminated due to a network failure, a disconnect notice is displayed on the terminal screen. The disconnect notice has four options: **Reconnect**, **Close**, **Cancel** and **Info**. If you click the **Reconnect** button, AccuTerm will attempt to reconnect to the server. If you click **Close**, AccuTerm will close the session. If you click **Cancel**, the reconnect notice is cleared (you can reconnect later by choosing

Tools ► Connect to Server from the main menu or by clicking the  button on the toolbar.) Clicking on the **Info** button will display a brief description of the network event that caused the disconnect.

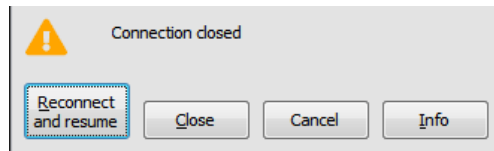


If your session is "resumable" (using AccuTerm's ReZume feature), the **Reconnect** button is renamed **Reconnect and Resume**. See [ReZume](#) for more details.

1.2.2 ReZume

AccuTerm 8 includes a unique feature called **ReZume**, designed to maintain the terminal session during a network outage, and resume the session once the network is restored.

When a "resumable" session is interrupted by a network outage, the first button on the **Reconnect** panel is changed from **Reconnect** to **Reconnect and Resume**. Clicking the **Reconnect and Resume** button will attempt to reconnect to the server, and once connected, resume the session exactly where it was interrupted.



ReZume requires a session manager be installed on the server, and some minor adjustments to the way the session is initiated on the server. Please see this [topic](#) on the web for instructions on configuring AccuTerm ReZume, installing the session manager on the server, and setting up a resumable session on the server.

Notes:

1. *ReZume is a premium feature, and requires an active maintenance subscription.*
2. *ReZume is currently only available for Secure Shell connections.*
3. *ReZume requires a companion session manager to be installed on the server. The session manager is available for Linux and AIX platforms.*

1.3 Product Activation

You can activate AccuTerm 8 immediately after the installation by clicking the **Activate** button from the Evaluation notice screen. You can also enter your activation code at any time by selecting **Activate** from the **Help** menu. Activation requires access to the Internet.

AccuTerm periodically contacts the license service over the Internet to validate your license. This ensures that premium features (which are subject to a valid maintenance subscription) remain functional without needing to update the license.

You can verify your activation status from the **Activate** screen (**Help ► Activate**) or from the **About** screen (**Help ► About AccuTerm**).

License type

AccuTerm 8 licenses are either user-based, or device-based. User-based licenses associate a unique license key for each user (email address). Users are allowed to activate up to two machines using their license key. Device-based licenses use a common license key for the entire organization. When activating AccuTerm, be sure to select the correct license type. If you are unsure, please contact your system administrator.

Email address

Enter the email address associated with your license key, unless you are using device-based licensing.

License key

Enter the license key you received by email or from your administrator.

Status

The status box shows the current status of the activation process. If an error occurs during activation, hover the mouse over this field to display the error code, which will be useful if you need to submit a request for support.

1.4 AccuTerm User Interface


Flexible Tabbed Interface

AccuTerm 8 has a new tabbed interface that supports multiple sessions in multiple main windows. This interface allows you to:

- Drag tabs on the session tab bar to re-order them
- Drag a tab outside of the AccuTerm window to open in a separate window
- Drag tabs from one AccuTerm window to another AccuTerm window

Title Bar

The title bar is a standard Windows element that contains the title of the application: AccuTerm. There is

an application icon  at the left side of the title bar, and standard Windows icons for minimize, maximize, restore and close at the right side.

Main Menu

The [main menu](#) is displayed immediately below the main window title bar. It is used to control the AccuTerm application. The main menu contains top-level menu items for **File**, **Edit**, **View**, **Tools**, **MultiValue**, **Window** and **Help**. Clicking one of the main menu items displays a sub-menu with items relating to the top-level item.

Toolbar

The Toolbar is shown below the main menu. The Toolbar contains buttons that duplicate some of the menu items, a kind of shortcut for the menu item. If you let the mouse hover over a toolbar button, a small description of the button's action will be displayed. You can hide or show the Toolbar from the [View menu](#).

Session Tabs

Session tabs are displayed at the top of each AccuTerm window, with a tab corresponding to each open session. Click on a session's tab to activate it. The session icon is displayed at the left side of the session's tab. When the mouse is over a tab, the tab shows a small close (X) button which can be used to close the session.

Each session tab provides a context menu (right-click menu) which can be used to activate, clone, rename or close the session.

The [Window menu](#) can also be used to activate a session.

Session Window

Terminal sessions are shown in the space between the toolbar and the status bar. Multiple tabbed sessions can be opened in this window. The active session normally occupies the entire session window space. However, it is possible to arrange the session windows in "tiles" where the session window space is divided into equally sized tiles. The sessions can be tiled vertically or horizontally. The tiles are not moveable or sizeable.

Status Bar

The Status Bar is displayed at the very bottom of the AccuTerm window. The Status Bar indicates various status information about the current (active) session, such as the keyboard state (locked or unlocked), the cursor position, screen size, terminal type, printer status, file capture status and connection status. You can hide or show the Status Bar from the [View menu](#).

1.4.1 Keyboard Commands

Next Session - CTRL+TAB

If you have multiple session tabs, press **CTRL+TAB** to activate the next session. The sessions are sequenced in the same order as the tabs. You can press **TAB** multiple times while continuing to hold the **CTRL** key to scan the sessions until you find the one you want to activate.

Previous Session - CTRL+SHIFT+TAB

If you have multiple session tabs, press **CTRL+SHIFT+TAB** to activate the previous session. The sessions are sequenced in the same order as the tabs. You can press **TAB** multiple times while continuing to hold the **CTRL** and **SHIFT** keys to scan the sessions until you find the one you want to activate.

Print Screen - ALT+G

Prints the terminal screen on the currently selected printer.

Toggle Slave Printer Mode - ALT+P

Toggle slave printer mode 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. *Note: this command changes pages in the terminal page memory, not the history (scrollback) buffer.*

Next Page - CTRL+PGDN

Displays the next terminal page. *Note: this command changes pages in the terminal page memory, not the history (scrollback) buffer.*

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 (Scrollback)

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

Besides using the keyboard to navigate the history buffer, you can also use the mouse wheel. Just scroll the mouse wheel up or down to scroll the history.

Note: in order to scan the history buffer, you must specify the number of history rows to remember in [Screen Settings](#).


1.5 AccuTerm Main Menu

As is common in most Windows applications, AccuTerm provides a menu and toolbar to control the program. The menu provides complete control, and the toolbar provides quick access to the most common functions. The menu functions are described in this topic, as well as the toolbar buttons.


File Menu

The AccuTerm **File** menu contains options for creating, opening, closing and saving sessions, and printer setup and control. Common menu options may also be performed by clicking a button on the toolbar.

File ► New

To create a new terminal session, select the **New** option or click the  button on the toolbar. The **Basic connection settings** panel is displayed to help you create and configure the new session. This is the same panel that is displayed when selecting the **Easy connect** option on the [Quick start](#) panel.


File ► Open...

To open an existing session configuration file, select the **Open** option or click the  button on the toolbar. This creates a new session with the settings that were saved in the selected configuration file. *Note: AccuTerm 8 session configuration files are in JSON format, and have a different file extension (.atsp) from the configuration files in previous versions of AccuTerm. It is recommended that you convert your older .atcf files to the new format. To convert, just open the old file, and use **File ► Save As** to save in the new format.*

File ► Close

To close the active session, select the **Close** option. If you have made any changes to the session settings, you will have the opportunity to save them in a session 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 **Save As** 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 **Save Layout** or **Save Layout As** options.

File ► Printer On

Selecting this option toggles the slave printer on and off. When the slave 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 the slave printer state.

File ► Print Screen

Selecting this menu option or clicking the  button option 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 [Printer Settings](#) for details.

Recent Files

Opens one of the most recently opened terminal session configuration or layout files. *Note: you can configure the number of items to show in the recent file list in the [General Settings](#).*

File ► Exit

Select this option to terminate AccuTerm.

Edit Menu

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


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 enclosed 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 (while holding the left button) and the window will automatically scroll in that direction.

You can also click the  button to perform this function.

An option is provided to remove the selection rectangle after a Copy operation - see [Clipboard Settings](#) for details.

Edit ► Paste

This command will send the clipboard contents to the host computer. See [Clipboard Settings](#) for line-end and paste-end options. Clicking the  button on the toolbar will also perform this function.

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.

Edit ► Select ► Screen

This command will select the entire screen. You will see a dotted rectangle around the selection.

Edit ► Select ► All

This command will select the entire screen and all history. You will see a dotted rectangle around the selection.

Edit ► Select ► None

This command will remove the selection rectangle from the screen.



Edit ► Clipboard Options...

Select this option to adjust clipboard options, such as line-end character. See [Clipboard Settings](#) for details.

View Menu

The AccuTerm **View** menu contains options to select screen size, display or hide the toolbar and status bar and session bar.

View ► Extended Screen Size

Toggles the screen size between Normal (usually 80 by 24) and Extended (usually 132 by 24). See [Screen Settings](#) for information on setting the number of columns and rows. The screen size may also be changed by clicking the  or  button.

View ► Toolbar

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

View ► Status Bar

The status bar displays information about the current session: keyboard state, cursor position, screen size, terminal type, printer status, capture status, connection type and connection status. Check this option to display the status bar at the bottom of AccuTerm's main window. Un-check this option if you do not want the status bar displayed.

View ► Function Bar

The Function Bar is used to display a row of function key buttons at the bottom of the session window. These buttons perform the same action as the twelve function keys on the keyboard. Un-check this option if you do not want the Function Bar displayed.


The first button in the Function Bar indicates the shift state of the buttons (Normal, Shift, Ctrl, Shift+Ctrl). When this button is clicked a small popup menu is displayed which will allow you to lock the shift state for the buttons in the Function Bar. Normally, the Function Bar buttons track the actual state of the **SHIFT** and **CTRL** keys.

The default function key button captions are simply F1 ... F12, however, the captions may be customized in [Keyboard Programming](#). Options to select the bar position (top or bottom) and display the key number in the corner of each button are available in [General Settings](#).


Tools Menu

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


Tools ► Connect to server...

This is a toggle to connect to or disconnect from the server. The  button on the toolbar can also be used to perform this action. The icon will be solid black when the connection is active and will be gray when the connection is inactive.

Tools ► 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.

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 ► Customize Menu...

Select this option to open the [AccuTerm Menu Designer](#), which is used to customize the menu, toolbar and status bar.


Tools ► Upload...

Selecting this option displays the [File Upload](#) dialog. The File Upload dialog is used to initiate a manual file upload and allows you to select files to upload as well the protocol to use and other options. *Note: to upload data to a MultiValue host which has the AccuTerm MultiValue host programs installed, select **Import Data** from the [MultiValue menu](#) instead.*

Tools ► Download...

Selecting this option displays the [File Download](#) dialog. The File Download dialog is used to initiate a manual file download and allows you to select the target directory as well as the protocol to use and other options. *Note: to download data from a MultiValue host which has the AccuTerm MultiValue host programs installed, select **Export Data** from the [MultiValue menu](#) instead.*

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 ► Macro ► Record New Macro

Select this menu item to begin recording a new macro. All menu and toolbar actions, including changing settings and preferences, are recorded in the macro. Also, keystrokes sent to the host and responses from the host are recorded in the macro. When a macro is being recorded, a small control window is shown in the lower-right corner of the screen. Use the control window to end, pause or resume macro recording.

Tools ► Macro ► Run Macro

This menu item opens a sub-menu which contains a list of recorded macros. Select the macro you want to run from this sub-menu. *Note: a shortcut key can be assigned to a recorded macro to simplify running the macro.*

Tools ► Macro ► Edit Macros...

This option opens the **AccuTerm Script** window with the **Macro** tab selected. Use this window to examine or modify the VBA code for your recorded macros.


Tools ► Script Editor...

This option opens the **AccuTerm Script** window with the **Main** script tab selected. Use this window to create, modify, run and debug VBA scripts.

MultiValue Menu


This menu contains all of the functions which are specifically designed for users of MultiValue systems.

MultiValue ► Export Data...

Select this option or click the  button on the toolbar to launch the AccuTerm File Transfer Wizard to guide you through the process of exporting data from your MultiValue database. This

option requires the AccuTerm MultiValue Host Programs be installed on your MultiValue host.

MultiValue ► Import Data...

Select this option or click the  button on the toolbar to launch the AccuTerm File Transfer Wizard to guide you through the process of importing data to your MultiValue database. This option requires the AccuTerm MultiValue host programs be installed on your MultiValue host.


Tools ► wED Editor

AccuTerm 8 includes a Windows based text and program editor designed specifically for MultiValue users. If you are connected to a MultiValue host, you can access the wED editor by

selecting this menu option or clicking the  button on the toolbar. *Note: the wED editor is also invoked by using the WED command from TCL.*


Tools ► GUI Designer

AccuTerm 8 includes a development tool to help programmers create **GUI Applications** for MultiValue systems. If you are connected to a MultiValue system, and are at a command prompt (TCL, ECL, jsh, etc.), you can access the GUI Designer by selecting this menu option

or clicking the  button on the toolbar. *Note, the GUI Designer is also invoked by using the GED command from TCL.*

Tools ► phiReport

AccuTerm 8 includes a MultiValue reporting tool to help you quickly create stunning reports from data in your MultiValue files. [phiReport](#) is a premium feature, and is only available with an active maintenance subscription. If you are connected to a MultiValue system, and you are at a command prompt (TCL, ECL, jsh, etc.), you can access phiReport by selecting this menu

option or clicking the  button on the toolbar. *Note: phiReport is also invoked by using the PHIMENU command from TCL.*

Tools ► Host Programs ► Install

Select this option to start the MultiValue Host Program Installer. The installer simplifies the installation of the AccuTerm host programs by determining the MultiValue platform and uploading the correct installation files to the host.

Tools ► Host Programs ► Update

Select this option to update the MultiValue Host Programs to the version supplied with your AccuTerm installation and to install optional libraries, samples and demo programs.

Tools ► Host Programs ► Account Setup

Select this option to enable accounts on the MultiValue host to use the AccuTerm Host Programs. This is required if you would like to access the file transfer functions, the wED Editor, GUI Designer or GUI runtime library from accounts other than the ACCUTERM account.

Tools ► Host Programs ► Configuration

Select this option to adjust MultiValue Host Program options.

Tools MultiValue Options...

Select this menu item to display the **AccuTerm MultiValue TCL Rules** dialog. Several of AccuTerm's MultiValue tools (File Transfer Wizard, wED Editor and GUI Designer) require a special MultiValue server utility to function. The server utility is typically initiated from "TCL", and settings in this dialog define how AccuTerm gains access to TCL and starts the server program.

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 ► Full Size

This option displays the active session tab using the full size of the AccuTerm window. When this option is selected, only a single session is visible in the AccuTerm window. To show multiple sessions, use one of the Tile options, or open multiple AccuTerm windows.


Window ► Tile Horizontally and **Window ► Tile Vertically**

These options divide the AccuTerm window into a number of tiles, and each session is displayed in its own tile. Tiles are not moveable or sizable (except when the AccuTerm window is resized, the tile sizes will be adjusted).

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

The Help menu contains functions to display online help, documentation and program information. It also provides a function to enter your product activation code. You can get context-sensitive help at any time by clicking the  button.

Help ► Contents

This menu item will open the online help to the Contents tab, where you can browse the topics to locate the information you need.

Help ► Index

This menu item will open the online help to the Index tab, where you can search help topics by keyword to locate the information you need.

Help ? Activate

Select this menu item to open the **AccuTerm 8 Activation** dialog to enter your activation code.

Help ► Documentation

This menu item will open the User Manual, GUI Manual, Scripting Language Reference (VBA) Manual, Release Notes, or the License Agreement.


Help ► About AccuTerm

This menu item will open the **About AccuTerm** dialog which displays the product copyright notice, serial number, release, license summary, and registration information. A link is provided to check for product updates.

See also:

[Keyboard Commands](#)

1.6 Configuring AccuTerm

To modify AccuTerm's settings, use **Tools ► Settings...** from the main menu or click the  button on the toolbar.

AccuTerm's settings are organized into *categories* which are shown in a nested tree structure, with similar categories as branches from a common parent category. The tree is displayed on the left pane of the **Settings** window. When you select a category from the tree by clicking on it, the settings for that category are displayed in the right pane.

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

Category:

- [General Options](#)
- [Files & Folders](#)
- [Connection](#)
 - [Telnet](#)
 - [Secure Shell](#)
 - [Private Key](#)
 - [SSL/Telnet](#)
 - [Serial Port](#) (Legacy device)
 - [Modem](#) (Legacy device)
- [Terminal](#)
 - [Screen](#)
 - [Colors](#)
 - [Attribute Colors](#)
 - [Palette](#)
 - [Background](#)
 - [Keyboard](#)
 - [Font & Character Set](#)
 - [Sounds](#)
- [Printer](#)
 - [Screen Print](#)
 - [Slave Printer](#)
- [Clipboard](#)
- [Mouse](#)
- [File Transfer](#)
- [MultiValue Rules](#)
- [Windows](#)

1.6.1 General Settings

The **General Options** are applicable to the overall operation of AccuTerm and are not associated with any particular session. Some of these settings allow you to re-activate features that have been previously disabled, like Serial Port support.

Open session shortcuts in a new windows

If this option is checked, and there is at least one active AccuTerm session, then double-clicking a .atasp session file will open the session in a new window. If the option is not checked then the session is opened as a new tab in the most recently used AccuTerm window.

Show "quick start" page at startup

Check this option to display the "What would you like to do?" dialog. This dialog gives you the option to open a recent connection, browse for an existing session file or configure a new session by selecting the "Easy connect" option. Note that selecting "Easy connect" gives you the same dialog as the File > New menu option.

Function key bar: at top of screen

Check this option to show the function key bar at the top of the session window. If this option is not checked, the bar will be displayed at the bottom of the window. *Note: you must check the **Function Bar** item in the **View** menu to show the function key bar.*

Function key bar: show key numbers

Check this option to show tiny key numbers in the corner of each button in the function key bar.

Warn before disconnecting

If this option is checked then when you attempt to close a session which is currently connected to the host (Dial-up, Telnet and Secure Shell connections), you will receive a warning that the session is still connected. You can then choose to cancel, leaving the session open, or continue closing the session. When this option is not checked you will not receive this warning.

bt

Close AccuTerm when all sessions closed

If this option is checked, when the last session is closed, the AccuTerm program will automatically close.

Remember Caps Lock and Scroll Lock state when activating

When this option is checked, AccuTerm remembers and restores the state of the CAPS LOCK and SCROLL LOCK keys when the program is activated. Each session maintains its own independent state for these keys. *Note: this option is often unreliable - we are investigating a resolution for this.*

Reset menu & toolbar to factory settings

This function applies to either the user's personal custom menu or the AccuTerm master menu. Resetting the master menu required administrator privileges and you will need to start AccuTerm using "run as administrator". You might need to reset the menu if your custom menu file becomes corrupted or is in a non-functional state.

Enable legacy device support (Serial port & Modem)

Checking this option gives you the ability to configure legacy Serial Port and/or Modem settings.

1.6.2 File & Folder Settings

Default folder for session files

This field specifies which folder to start in when opening or saving a session file. Once you have opened or saved a session file, open or save will start with the last folder used. The default folder for session files is the current user's My Documents folder.

Default upload/download folder

This field contains the default folder for uploading and downloading files. You can click the **Browse...** button to browse for a folder.

Default capture folder

This field contains the default folder for storing captured data files. You can click the **Browse...** button to browse for a folder.

Recent file list size

Select the number of recent files displayed in the **File** menu. The default is 4 files.

Explore application data folder

Click this button to open a Windows Explorer window at the top of the AccuTerm application data folder. This folder is the user's profile under the AppData folder. This is the folder where the phone book file, Secure Shell keys and personal custom menus are stored.

1.6.3 Connection Settings

Connection type

The **Connection type** drop-down list is used to select a connection type. Based on the connection type selected, other configuration options may be available. For example, if the connection type is a network protocol, the **device licensing** options are available. The specific settings for each connection type will be described later in related topics.

Echo (duplex)

There are three echo options: remote echo (full duplex), local echo (half duplex) and loopback (local mode). Typical terminal sessions operate in full duplex. In this mode, all characters typed on the keyboard are sent to the host computer. They are not printed on the screen. The host echoes the character back to the terminal where it is displayed on the screen. Usually, when control characters (**ENTER**, **BACKSPACE**, **DELETE**, etc.) or function keys are sent to the host, the echo is suppressed and the application responds in an appropriate manner. In half duplex mode, characters typed on the keyboard are simultaneously sent to the host computer and displayed on the screen. In loopback mode, characters are displayed directly on the screen and are not sent to the host computer. This mode is usually used only for testing keyboard or display functions.

Clear high bit of received data

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 terminal types have no way of handling characters above 127 (hex 7F). In this case, you can check this box and the high 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 should not use this option (the Wyse 60 and VT emulations have commands to access these characters when using only 7 bits).

Enable device licensing (U2 / QM / D3)

If this option is set, multiple connections (sessions) to compatible versions of UniData, UniVerse, QM and D3 consume only a single user license. When the **Append LF** option is checked, the response sent to the host by AccuTerm is terminated by a line-feed character. Contact your U2, QM or D3 dealer for more information regarding device licensing. *Note: Windows versions of U2 implement device licensing using a proprietary Telnet server. This precludes using SSH with device licensing on these platforms.*

Display connection warnings

Check this option to display non-fatal connection errors, such as UART overrun errors when using a serial port connection.

Close session on disconnect

When this option is checked, if a connection is closed by the host computer the AccuTerm session will automatically close. This option may be used in conjunction with the **Close AccuTerm after last session is closed** option ([General Options](#)).

Notify server when number of columns or rows change

Check this option to send a "screen resize" (SIGWINCH) signal to the server when the number of columns or rows change. This is needed when the "Variable" screen size mode ([programmer mode](#)) is selected in the screen settings. It may also be used to notify the server when the toolbar button is used to toggle the screen between 80 column (normal) and 132 column (extended) mode.

Connection settings

Click the **Connection settings** button to open the settings panel for the selected connection type.

ReZume (Session Resilience) settings

The **ReZume** button is available when the connection type is **Secure Shell**, and when an AccuTerm 8 Maintenance subscription is active (**ReZume** is a premium feature). Click this button to open the **ReZume** settings panel.

See also:

[Telnet Settings](#)

[Secure Shell Settings](#)

[SSL/Telnet Settings](#)

[ReZume Settings](#)

1.6.3.1 Device Settings:Telnet

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

Host name or IP address

Enter the host name or IP address in this field. If the desired host name has been used previously when configuring a Telnet session, you may select it from the drop-down list by clicking the drop-down arrow in this field.

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 configured to be compatible with the specified host type. For some host types, the port number and host terminal type are also adjusted.

Port

If the Port field is left blank, or set to "Default" AccuTerm will use TCP port 23 (the default port for Telnet). If a different port is used for Telnet connections to your host, enter that port number in this field.

Certain host systems, such as Raining Data D3, allow a host process number (PIB) to be mapped to a specific TCP port number. You can use the 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.

Note: this timeout only affects the time AccuTerm waits for a response from the host when opening a connection to the host.

Advanced Options

Check the **Show advanced options** check-box to display the advanced Telnet options.

Binary

Normally, Telnet operates in "text mode", with special rules for handling *CR* and *LF* characters. When the **Binary** option is selected, Telnet sends *CR* and *LF* characters without any alteration. *Some hosts do not function correctly in binary mode.*

Keepalive

Although the Telnet protocol specification does not provide for "keepalives", AccuTerm can send periodic commands that should be ignored by most Telnet servers. Using keepalives may prevent the host (or router) from detecting that the connection is idle and forcing a disconnect. To use keepalives, select either DMARK or NOP as the command to send as a keepalive. Enter the desired interval in the **every... sec** field.

Break key

Normally, AccuTerm sends the Telnet "interrupt process" command when you press the **CTRL+BREAK** key. If your host requires a different BREAK command, or to use a control character instead of a command, select the desired option from this drop-down list. If you select "control character", enter the decimal value of the control character to send in the **code** field (for **CTRL+C**, the decimal value is 3).

Skip initial option negotiation

When a Telnet connection is first established the AccuTerm and the host negotiate various operating modes. When this option is selected, the initial option negotiation is suppressed. This is necessary for some Telnet servers which do not respond correctly to Telnet options (Advanced Pick and D3 turnkey or nailed Telnet).

Disable outbound coalescing

Normally, TCP delays outbound messages for a short time so that multiple small messages may be combined into a single network packet. This prevents many small packets from clogging up the network. When this option is selected, messages are sent as soon as possible. This may improve throughput (especially when running GUI programs or performing file transfers), at the expense of increased network traffic.

Allow internationalized domain names

Select this option to enable Internationalized Domain Names (IDN) for the host name. IDN relies upon a standardized mechanism known as "Punycode" for encoding Unicode domain names using only the ASCII characters that are permitted by the global DNS system.

1.6.3.2 Device Settings: SSH

AccuTerm supports Secure Shell (SSH) connections to host machines over a TCP/IP network connection. Any SSH server on the network, including hosts on the Internet, can be accessed. AccuTerm implements both version 1 and version 2 of the Secure Shell protocol. Protocol selection is usually automatic and uses version 2 if it is supported by the host. Otherwise, it automatically falls back to version 1.

AccuTerm Secure Shell requires Windows 128 bit encryption. You can check your encryption capabilities by opening Internet Explorer and selecting "About Internet Explorer" from the Help menu. If the Cipher Strength is less than 128 bits you will need to either upgrade to Internet Explorer 6 (or above), or install the High Encryption Pack for your version of Internet Explorer. Search Microsoft's site for "High Encryption" for more information.

Host name or IP address

Enter the host name or IP address in this field. If the desired host name has been used previously when configuring a Telnet session, you may select it from the drop-down list by clicking the drop-down arrow in this field.

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 configured to be compatible with the specified host type. For some host types, the port number and host terminal type are also adjusted.

Port

If the Port field is left blank, or set to "Default" AccuTerm will use TCP port 22 (the default port for SSH). If a different port is used for SSH on your host, enter that port number in this field.

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.

SSH version

Select the version of the SSH protocol that will be used by the connection. When you select "auto", AccuTerm will automatically select the highest version supported by the host. If you select SSH1 or SSH2, AccuTerm will only connect using the selected protocol. If the host does not support the selected protocol, the connection will fail and an error message will be displayed.

Authentication

Select the authentication method which will be used to validate the user with the host. AccuTerm supports "password", "interactive" and "public key" authentication. It is important to select an authentication method that is compatible with the SSH server running on your host.

Private key

If "public key" authentication has been selected, click the **Select...** button to open the [Key Management](#) dialog. Use the [Key Management](#) dialog to create new keys, load existing keys and import keys created on another system.

Host validation

When an SSH connection is established, the host sends its public key (DSA or RSA) to AccuTerm. AccuTerm supports various host key validation levels:

None: the host key is ignored (this means that you trust that the host you connect to is the desired host).

Trust on first use: if the key has not been established (first time you connect to a particular host), AccuTerm assumes you are connected to the desired host and its key is saved in your session configuration. The key is verified on subsequent connections. If verification fails, the connection is aborted.

Confirm change: a dialog is displayed if a change in the host key for a particular session is detected. Confirmation is required to complete the connection. If the host is confirmed, the new key is saved in the session configuration. Otherwise, the connection is aborted.

Verify host key fingerprint: verifies that the fingerprint in the Host Key Fingerprint field matches the key supplied by the host. If verification fails, the connection is aborted.

Host key fingerprint

This is the host key fingerprint used for verifying the host key. The fingerprint is the MD5 or SHA1 hash of the host RSA or DSA public key, in hexadecimal form. If your host is running OpenSSH, you can use the ssh-keygen command to display the host's public key fingerprint. The command line syntax is: `ssh-keygen -lf /etc/ssh/ssh_host_rsa_key.pub` (change "rsa" to "dsa" to display the DSA key

fingerprint instead). The `ssh-keygen` command displays the MD5 hash of the host key in hexadecimal, with the key length shown before the hash. Only the hash (not the key length) should be entered into the Host key fingerprint field. *Note: when running AccuTerm in FIPS-140 mode, you must use the SHA1 hash because the MD5 hash algorithm is not supported in FIPS-140 mode.*

Advanced Options

Check the **Show advanced options** check-box to display the advanced Telnet options.

Preferred cipher

If you wish to select the cipher used to encrypt the SSH session, you can select it from this list. If "auto" is selected, the default cipher is used. Otherwise, you can choose Triple DES, Blowfish, 128 bit AES or 256 bit AES. The default cipher for SSH1 is Triple DES. The default cipher for SSH2 is 128 bit AES in CTR mode (aes128-ctr).

Only use preferred cipher

When this option is set, the cipher selected above must be supported by the host. If the preferred cipher is not supported by the host, the connection will be terminated and an error message displayed. When this option is not set, if the preferred cipher is not supported by the host, a different, mutually supported, cipher (normally, Triple DES) will be used.

Break key

Normally, AccuTerm sends `CTRL+C` when you press the `CTRL+BREAK` key. Alternatively, you can opt to send the `BREAK` command or `INT` signal when the `CTRL+BREAK` key is pressed. In order to use the `BREAK` command, the SSH server on your host must implement the Session Channel Break Extension spec (RFC 4335). In order to use the `INT` signal, the SSH server must implement signals, and must respond to the `INT` signal by interrupting or breaking the process. Not all SSH servers implement these features. To use a control character instead of a command or signal, select "control character" from this drop-down list and enter the decimal value of the control character to send in the **code** field.

Send keepalives

When the Send keepalives option is selected, AccuTerm sends a dummy message to the host periodically. This may prevent the host (or router) from detecting that the connection is idle and forcing a disconnect. Enter the desired interval in the **every...sec** field.

Disable outbound coalescing (advanced option)

Normally, TCP delays outbound messages for a short time so that multiple small messages may be combined into a single network packet. This prevents many small packets from clogging up the network. When this option is selected, messages are sent as soon as possible. This may improve throughput (especially when running GUI programs or performing file transfers), at the expense of increased network traffic.

Allow internationalized domain names

Select this option to enable Internationalized Domain Names (IDN) for the host name. IDN relies upon a standardized mechanism known as "Punycode" for encoding Unicode domain names using only the ASCII characters that are permitted by the global DNS system.

1.6.3.2.1 Key Management

Use the **Key Management** dialog to create public/private keys for SSH, or to load or import private keys. AccuTerm supports RSA public keys for both SSH 1 and 2, and DSA public keys for SSH 2. AccuTerm can import SSH 2 private keys that were created by OpenSSH or `putty` (`puttygen`). AccuTerm can save public/private key files in either OpenSSH (www.openssh.org) or `putty` format.

Creating a new key

Click the **Create New Key** button to create a new public/private key pair. After clicking the **Create New Key** button, select the key type (RSA or DSA), and key length. The default length of 1024 bits is recommended. Enter a descriptive comment describing the key being created, and enter a password to protect your private key. Although the password is optional, it is highly recommended since anyone who gains access to your private key file can impersonate your identity! The **Save Key** button will remain disabled until the password has been confirmed.

Click the **Save Key File** button to create the new key and save the newly created key file. By default, AccuTerm will save the key in its standard key folder, located in the user's profile. You can select a different location, such as a removable USB drive, if desired.

Once your public/private key pair has been created, you need to place the public key on the server which you want to connect to. Use the **Export Key** or **Copy Key to Clipboard** functions to retrieve your public key so that it can be installed on the server.

Exporting the key

You can save the public key to a file by clicking the **Export Key** button and selecting the **Public Key** option. Click the **Export** button to save the public key file. Browse to the folder where you want to store the public key file and click the **Save** button. Once saved, you could attach the key to an email message and send it to the server's administrator (it is a public key, so security is not a major issue). You could also save the public key file to a removable drive or diskette and deliver it to the server administrator.

The same process can be used to export the private key file. *Note: exporting the private key may pose a security risk. If an unauthorized user obtains your private key he can impersonate you and gain access to the host system using your identity! Use caution when making a copy of your private key.*

Copying the key to the clipboard

Click the **Copy Key to Clipboard** button to copy either the public key (recommended) or private key (not recommended) to the clipboard. The process is the same as described under **Exporting the key** except that you do not need to supply a destination file.

Loading an existing key

If your private key was saved to a different folder or to a removable drive or diskette, you can click the **Load Existing Key** button to load that key. Enter the password and click the **Verify** button to check the password. If the password is correct, the key fingerprint will be displayed. Click **OK** to use this key (in its current location). Otherwise, you can save the key in a different location by clicking the **Save Key** button.

Importing a key

Importing a key is exactly like loading an existing key, except for the file locations. AccuTerm can import private keys created by OpenSSH.

Key fingerprint

When you create a new key, or load or import an existing key, AccuTerm displays the "fingerprint" of the public key. You can use this fingerprint to ensure that your key is correct. Use a utility on the server to show the fingerprint of the public key and compare to the fingerprint shown in AccuTerm.

OpenSSH example

If the host is using OpenSSH as the SSH server, and if you currently have access to the server via password authentication, you can set up public key authentication using the following steps:

1. Open an AccuTerm session and connect to the server. Log in using your password.
2. Open the AccuTerm Settings dialog to create your new key pair. Select "public key" as the authentication method.
3. Click the **Select Key** button to open the **Key Management** dialog.
4. Click the **Create New Key** button and follow the steps above for creating a new key.
5. Click the **Copy Key to Clipboard** button and choose the **Public Key** option. Click the **Copy** button.
6. Click **OK** two times to dismiss the dialogs. Ignore the warning about changes not being effective.
7. From a Unix/Linux prompt, change to your `~/.ssh` directory. Edit the `authorized_keys2` file using your favorite text editor. Start a new line in the `authorized_keys2` file.
8. From the AccuTerm **Edit** menu, choose **Paste** to paste the public key into the `authorized_keys2` file.
9. Save the modified `authorized_keys2` file.
10. Save the modified AccuTerm session file. Close AccuTerm. Open AccuTerm and open the session file you just modified.
11. Enter your user ID.
12. If you entered a password when you created your new key, you will prompted to enter the password. Enter the password (case sensitive).
13. If the server accepts your key, you will be logged in.

1.6.3.3 Device Settings: SSL/Telnet

AccuTerm 8 supports SSL/Telnet connections to host machines over a TCP/IP network connection. This connection method is used primarily for hosts running on Windows which do not normally support Secure Shell. This connection method establishes a secure connection using industry-standard SSL/TLS, then initiates a telnet session within the secure connection (tunnel). UniVerse and UniData platforms running under Windows have built-in support for this connection method. Other platforms may use an SSL proxy such as *stunnel* to support this connection method.

Many of the SSL/Telnet settings are shared with the normal [Telnet settings](#). Please refer to the Telnet settings for details. Only the settings specific to SSL/Telnet are described in this topic.

Host name or IP address

Enter the host name or IP address in this field. If the desired host name has been used previously when configuring an SSL/Telnet session, you may select it from the drop-down list by clicking the drop-down arrow in this field. *Note: depending on how you configure Server Validation, the value in this field may need to match the Common Name attribute of the host's certificate.*

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 configured to be compatible with the specified host type. For some host types, the port number and host terminal type are also adjusted.

Port

If the Port field is left blank, or set to "Default" AccuTerm will use TCP port 992 (the default port for SSL/Telnet). If a different port is used for SSL/Telnet connections to your host, enter that port number in this field.

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.

Note: this timeout only affects the time AccuTerm waits for a response from the host when opening a connection to the host.

Advanced Options

Check the **Show advanced options** check-box to display the advanced SSL and Telnet options.

Server Validation

When establishing an SSL/TLS connection, the host will send its *server certificate* to AccuTerm, so that AccuTerm can authenticate the host's identity. AccuTerm supports a variety of validation options:

None: any server certificate is assumed to be valid.

Minimal: the server certificate must not have been revoked. The certificate's name, role, expiration and root certificate authority are ignored.

Self-signed - disregard host name: the server certificate must not be expired and must not be revoked. The certificate's name, role and root certificate authority are ignored.

Self-signed - verify host name: the server certificate's Common Name must match the value in the "Host name or IP Address" field (above). The certificate must not be expired or revoked. The certificate's role and root certificate authority are ignored.

Strict - the server certificate must be issued by a trusted certificate authority, and it must be valid for use as a Server certificate. It must not be expired or revoked, and the Common Name must match the value in the "Host Name or IP Address" field (above).

Custom - click the Custom button to customize the validation options.

Client certificate

The Client certificate is used for client authentication when establishing an SSL/TLS connection. Client certificates are not normally used. If the server requires a client certificate, select the desired certificate from the drop-down list. Only certificates installed in the Windows Certificate Store are shown in this list. *Note: for UniVerse and UniData, the "peer names" are not case-sensitive.*

Advanced Telnet Options

See [this](#) topic for a description of the advanced Telnet options.

1.6.3.4 Device Settings: Serial Port

While AccuTerm 8 continues to support serial ports, this is a legacy connection and will not be shown as an available connection type unless the "Enable legacy device support" option is checked in the General settings

Port

AccuTerm supports serial ports from COM1 to COM127, at baud rates up to 115,000 baud. Although you can select any port from 1 to 127, most computers only have one or two ports installed, and some may not have any. The drop-down list contains all of the ports that AccuTerm was able to identify on your computer.

Baud rate

Baud rate is the speed that the serial port communicates with the host computer. 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 19200 baud. The speed that you use depends on the host computer. If you select the wrong baud rate, your screen may appear garbled or you may get no response from the host when you type on the keyboard.

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.

Flow control (handshake)

Computers can send information very quickly. However, there are situations where one computer can send data faster than the other computer can process it. In these cases, the computer on the receiving end needs to inform the sending computer to suspend transmission until the slower computer is able to catch up. A technique called "handshaking" is used to accomplish this task.

AccuTerm supports three methods of handshaking. The first and most common is. Xon/Xoff (or *software handshake*). 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 the Xon character (DC1). AccuTerm normally uses "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. If you check the "bidirectional" option, then AccuTerm will use both "inbound" and "outbound" Xon/Xoff handshake allowing the host to control the flow of data from AccuTerm as well.

The second handshake method supported by AccuTerm is DTR/DSR. This method is referred to as hardware handshake because it physically changes the voltages on the serial port to indicate whether the receiving computer is ready.

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 DTR/DSR or RTS/CTS handshake 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 reference.

This setting must match your host system.

Break key

Normally, AccuTerm sends a BREAK signal when you press the CTRL+BREAK key. Alternatively, you can use a control character instead of the BREAK signal. If you select "control character", enter the decimal value of the control character to send in the **code** field (for CTRL+C, the decimal value is 3).

1.6.3.5 Device Settings: Modem

While AccuTerm 8 continues to support modems, this is a legacy connection and will not be shown as an available connection type unless the "Enable legacy device support" option is checked in the General settings

AccuTerm 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 connections are initiated (dialed) entering the destination phone number in the Dial dialog, and clicking the **Dial** button. Connections are terminated by clicking the **Hang Up** button in the Dial dialog.

The Dial dialog is opened when you click the **Connect to server** button  on the toolbar.

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" option.

Dialing Options

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

Override settings from Control Panel

When this option not selected, the modem settings (baud rate, data bits, stop bits, parity and handshake) are determined by the modem properties set in Control Panel (Phone and Modem Options). Select this option to override the Control Panel settings.

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 19200 baud. The speed that you use depends on the host computer.

Only connect at this speed

Select this option to lock the modem speed to the selected baud rate. When this option is not selected the specified baud rate is the maximum connection speed. *Note: not all modems support this option.*

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.

Flow control (handshake)

AccuTerm supports two methods of handshaking for modem connections: software (Xon/Xoff) or hardware (RTS/CTS). Most modern modems work best with hardware (RTS/CTS) handshake.

Break key

Normally, AccuTerm sends a BREAK signal when you press the **CTRL+BREAK** key. Alternatively, you can

use a control character instead of the BREAK signal. If you select "control character", enter the decimal value of the control character to send in the **code** field (for CTRL+C, the decimal value is 3).

1.6.3.6 ReZume Settings

Allow session to reconnect and resume after a network outage

Check this box to enable **ReZume** for the session. See this [topic](#) for more information about **ReZume**.

After resuming, resend lost data up to:

During a "resumable" session, AccuTerm will save the most recent data which was sent to the host. When a session is "resumed", any data lost during the network outage can be re-sent. This field is used to specify the size of the buffer AccuTerm will use to save the most recent data. If you do not want to resend lost data when resuming, set this to 0.

Note: AccuTerm 8 ReZume is a premium feature, and requires an active maintenance subscription.

1.6.4 Terminal Settings

Terminal Emulation

In order for AccuTerm to work properly with terminal-based applications, the terminal type that AccuTerm is emulating must match the terminal type the host application is configured or designed to use. Select the terminal type (emulation or personality) from the drop-down list. If you select the wrong type, your screen will probably be scrambled, and you should try a different type. The following terminal types are supported by AccuTerm 8:

Wyse 50: emulates a Wyse 50 terminal with "Enhanced" enabled. Wyse 50 emulation may also be used for Wyse 30, 99, 125 and 350 terminals.

Wyse 60: emulates a Wyse 60 terminal with "Enhanced" enabled. Wyse 60 emulation is similar to Wyse 50, but with non-embedded visual attributes (the visual attribute codes do not take a position on the screen). Wyse 60 emulation may also be used for Wyse 150, 160 and 320 terminals.

ADD5 Viewpoint A2: emulates the ADD5 Viewpoint A2 terminal. This is the original A2 terminal, with no enhancements.

ADD5 Viewpoint 60: emulates the ADD5 Viewpoint 60 terminal. The Viewpoint 60 is similar to the A2, but uses embedded visual attributes (the visual attribute codes take a position on the screen).

Procomm VP60: emulates an ADD5 Viewpoint 60 terminal, but with non-embedded visual attributes. This emulation is similar to the Viewpoint 60 emulation provided by Procomm.

Viewpoint A2 Enhanced: emulates a Wyse 50 terminal in ADD5 Viewpoint mode with "Enhanced" enabled. This is basically a hybrid of Wyse 50 and ADD5 Viewpoint A2. Viewpoint A2 Enhanced may also be used for ADD5 2020 and 4000, and NCR 2900 terminals.

MDIS P30 (Prism): emulates a McDonnell Douglas Prism P30 terminal.

VT52: emulates the DEC VT52 terminal.

VT100: emulates the DEC VT100 terminal. VT100 is a 7-bit emulation, and can also be used for applications which require VT102 terminals.

VT220: emulates the DEC VT220 terminal. The VT220 emulation supports a super-set of the VT100 commands, and includes a host-writable status line, protected characters, multiple character sets and multiple pages.

VT320: emulates the DEC VT320 terminal. The VT320 emulation supports a super-set of the VT220 commands, and includes support for reporting and restoring various terminal states.

VT420: emulates the DEC VT420 terminal. The VT420 emulation supports a super-set of the VT320 commands, and includes support for rectangle operations, left & right margins, and horizontal split screen.

ANSI BBS: emulates the old MS-DOS ANSI.SYS device driver. This emulation may be useful when connecting to bulletin board systems.

Linux Console: emulates the Linux console, including function keys and colors.

SCO Console: emulates the SCO Unix console.

Pick PC Monitor: emulates the original Pick PC Monitor ("mm-mon" or "I" terminal type), including color and function keys.

Tektronix 4014: emulates the Tektronix 4010/4014 terminals. Color enhancements as implemented in the CIT467 terminal (from CIE Electronics) are included.

Tektronix 4100: partially emulates the Tektronix 41xx terminals.

TTY Emulation: emulates basic TTY functions.

Host terminal type

When a network connection is initiated, the contents of the **Host terminal type** field is passed to the host. The host may use this information to set its TERM environment variable or setting. Generally, when you change the **Terminal emulation** setting, the **Host terminal type** is automatically updated. The two fields are similar, but normally are not identical. The **Host type** setting ([Telnet](#) or [Secure Shell](#)) may influence the **Host terminal type** setting.

Answerback

The contents of this field are sent to the host when the application sends a "send answerback" command to AccuTerm. Wyse and VT emulations support the answerback command. When the answerback message is sent to the host under Wyse emulation, an ACK control character follows the message. Under VT emulation, no terminator is sent.

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

Terminal Options

Certain terminal types support additional settings shown in the **Terminal Options** panel.

Send 8 bit controls

If this option is selected, then 8-bit control codes will be transmitted to the host. If the option is not set (the default), equivalent 7-bit escape sequences will be transmitted to the host. This option applies to the VT220, VT320, VT420 and Linux Console emulations.

Allow application mode

When this option is selected (the default), commands that switch the numeric keypad keys and cursor keys into "application mode" are emulated. When the option is not selected, those commands are ignored, preventing the terminal from switching to "application mode". This option is useful when the host or application erroneously sends the command to enter application mode, and the application is expecting normal numeric or cursor keys. This option applies to the VT emulations (except VT52) and Linux Console emulation.

Keypad sends application codes

When this option is selected, the numeric keypad keys send special application codes. When the option is not set (the default), the numeric keypad keys send decimal numbers. This option applies to the VT emulations (except VT52) and Linux Console emulation.

Cursor keys send application codes

When this option is selected, the cursor keys send special application codes. When the option is not set (the default), the cursor keys send normal cursor codes. This option applies to the VT emulations (except VT52) and Linux Console emulation.

Backspace key sends DEL

When this option is selected, the **BACKSPACE** key sends the *DEL* (delete) control code. When the option is not set (the default), the **BACKSPACE** key sends the *BS* (backspace) control code. This option applies to the VT emulations (except VT52) and Linux Console emulation.

Auto-print works just like VT terminal

When this option is selected, the "auto print" slave printer function works the same as a real VT terminal. That is, when AccuTerm is in "auto print" mode, the entire screen line that the cursor is on is printed when AccuTerm receives a *CR*, *LF* or *FF* control code. If this option is not set, then text is printed as it is received from the host (similar to Wyse auto-print mode).

Advanced Options**Environment variable**

When a network connection is initiated, AccuTerm can send a user-defined environment variable to the server. Enter the name and value of the environment variable. Names should be appropriate environment variable names, and values may not contain = or , characters. Values can be null. This feature can be used to send identification information, like port number, to the server. *Note: the server may ignore environment variables sent by AccuTerm. The server's SSH configuration normally requires listing trusted environment variables in the AcceptEnv section of sshd_config.*

1.6.4.1 Screen Settings

Use the **Screen Settings** panel to specify the size of the terminal screen, the number of screen pages, the number of history rows, the status line and other options.

Screen Size

AccuTerm supports three screen size modes: Normal, Extended, Variable (programmer mode)

Normal and Extended are used for the 80 and 132 column modes of the Wyse and VT terminals. AccuTerm has taken these modes one step further in that you can define both modes to be any number of columns or rows, up to a maximum of 240. You can select which screen size is active when the session is first opened by checking the **Active** box for the desired size. You can leave these at their default settings of 80 columns by 24 rows for Normal mode and 132 columns by 24 rows in Extended mode or change them to accommodate applications that require a different screen size.

Note that the number of rows may change if you change terminal emulations. The console terminal emulations use 25 rows with no status line; others use 24 rows with one or more status lines.

When AccuTerm switches between the normal and extended modes, one of two things will happen: either the font 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 font size (see [Fonts](#)).

The new Variable mode (programmer mode) will adjust the screen size (columns and rows) to fit the window size. When Variable mode is selected, AccuTerm will send a "screen resize" signal ([SIGWINCH](#)) to the server when the window is resized. Applications that respond to the "screen resize" signal, like vi or the jBASE jed command, should automatically adjust the screen for the new size.

Screen pages

Some of the terminals which AccuTerm emulates support multiple addressable pages. With AccuTerm, you can have up to 25 pages.

History rows

AccuTerm supports a "history" or "scrollback" buffer, which saves lines which have scrolled off of the terminal screen. To use this feature, specify how many rows (up to 10,000) 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.

Status & message lines

Some of the terminals that AccuTerm can emulate display status, function key labels or host messages on the top or bottom lines of the screen. Depending on the terminal emulation selected, you may be able to adjust the number of status lines to use.

For Wyse and ADDS emulations, up to 3 status lines are available: a host message line at the top of the screen, and a "shifted" label line and "unshifted" label line at the bottom of the screen. For VT emulations, only a single status line is available for displaying a message from the host at the bottom of the screen. For the console emulations, the status line is not supported.

No message or status line: select if you do not want any status or message lines.

Shared message & status line: select to use a single, shared line at the bottom of the screen. If the native terminal supports more than one line (Wyse and ADDS Viewpoint), the bottom row of the screen displays the most recently used line. The line is shared between label line, shifted label line and host message line.

Split message & status lines: select if you want two separate status lines. The host message line is the top row of the screen display, and the label line is displayed on the bottom row. If the shifted label line is programmed, pressing the SHIFT key displays the shifted label line. Releasing the SHIFT key displays the unshifted label line.

Separate message, status & shift lines: select if you want three separate status lines. The host message line is the top row of the screen display, and the shifted and unshifted label lines are displayed on the bottom two rows.

Show scrollbar

Select this option if you would like a vertical scroll bar at the right edge of the screen window. The vertical scroll bar is useful when scanning history, or if the window is too small to display the complete screen.

Allow Tektronix graphics

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

emulation mode when graphics mode is terminated.

Automatic line wrap

When this option is checked, automatic line-wrap is enabled. When the cursor reaches the end of a line, typing another character normally causes the cursor to move to the beginning of the next line. When this option is not set, the cursor does not move.

Cursor style

AccuTerm supports either a blinking block or blinking underscore for the terminal screen cursor.

Protected attribute

Wyse 50 and Wyse 60 terminals use a visual attribute to identify "protected" characters. To change the "protected" attribute, click the drop-down arrow and select the desired attribute.

1.6.4.1.1 Color Settings

Most of the terminals which AccuTerm emulates do not support color. Instead, they have visual attributes like normal, reverse, dim, blink, underline, etc. While AccuTerm supports these visual attributes, it also supports color, even for terminals without native color support. This is done by assigning a foreground (text) and background color to each of the visual attributes that the native terminal supports.

Besides the visual attributes and colors, AccuTerm also supports "Visual Styles". This is a feature that automatically surrounds rectangular blocks on the terminal screen with a thin border. The border resembles the border that Windows uses for text fields, lists, buttons and frames. Effective use of Visual Styles can make an ordinary text screen appear very Windows-like.

Preview

The **Preview** window displays a sample screen with commonly used attributes using the current color settings. To choose a different color for any attribute in the preview window, click on the name of the attribute you would like to change. A drop-down list showing all possible color combinations will be displayed. Click an item in the list to change the foreground and background color of the attribute.

Theme

The **Theme** buttons provide an easy way to select one of the built-in color themes. Themes consist of the palette colors, attribute colors and Visual Styles. To preview any of the built-in themes, hover the mouse over one of the theme buttons. The **Preview** panel temporarily changes to the theme under the mouse so you can see the effect without actually changing the current settings. To select one of the built-in themes, click the theme's button.

Some of the built-in themes utilize AccuTerm's "Visual Styles" feature which can make character-based screens look like Windows. This effect requires careful use of the visual attributes by the application. Each rectangular block of characters on the screen with a common visual attribute will be enclosed in a thin border, if one is defined for that attribute. Visual Styles supports 3 border styles: inset, raised and flat.

The built-in themes are:

Default colors: uses a palette of 8 dark and 8 light colors. The dark colors are a bit darker than the dark legacy colors and the light colors are a bit lighter. The "normal" color (this is the color that the screen displays any time it is cleared) is light gray with black text.

Legacy colors: uses a palette of 8 dark and 8 light colors. The palette is essentially the same as

the original VGA palette. This is the palette that all prior versions of AccuTerm have used. The "normal" color is dark blue with yellow text.

VS Code Dark: uses a palette and color scheme that resembles the "dark" theme of Visual Studio Code.

White on Black: resets the palette to the 16 default palette colors and assigns combinations of black, light gray, dark gray and black to each of the visual attributes. The normal color is black with light gray text.

Black on White: similar to White on Black, except that the normal color is light gray with black text.

Green Screen: resets the palette to the default palette, except that the Olive color is a medium shade of green. Assigns combinations of light green, medium green, dark green and black to each of the visual attributes. The normal color is black with medium green text.

Reverse Green Screen: similar to Green Screen, except that the normal color is medium green with black text.

System colors: sets several colors in the palette to "system colors" defined in the Windows Display Properties. The normal background color is "Window background" and the normal text color is "Window text." The following palette colors are replaced by a system color:

- Black = Window Text Color
- White = Window Background Color
- Dark Blue = Highlight Background Color
- Olive = Highlight Text Color
- Light Gray = 3D Light
- Dark Gray = 3D Shadow

Other palette colors are set the same as the default colors.

Classic Windows: this theme enables AccuTerm's Visual Styles. It uses the same palette as System Colors and assigns an inset border style to the "reverse" and "dim reverse" attributes, so that text displayed using these attributes resembles a typical Windows text field. A raised border is assigned to "underline" and "dim underline" which resembles a Windows command button. The normal background color is "3D face" and the normal text color is "Window text."

Modern Windows: this theme is similar to Classic Windows except that a flat border is assigned to "reverse" and "dim reverse". This is similar to the border that Windows XP and later uses for text fields.

Advanced Colors & Styles

Click this button to open the [Attribute Colors](#) panel, which allows you to fine-tune the colors and border effects for each visual attribute.

Border Style

If you are using **Visual Styles** (or the AccuTerm private programming command to draw borders), you must select a border style. Choose "thin" for a one-pixel border, or "thick" for a two-pixel border. When

external borders are selected, vertical border lines are drawn in adjacent character cells. Otherwise they are drawn inside the character cell and sometimes overlap the left or right edge of the character; thin borders work best with smaller font sizes, but are less dramatic than thick borders.

Line Drawing

AccuTerm supports the line and box drawing characters used by the native terminals. AccuTerm supports three line drawing styles: normal, flat and beveled. Choose "normal" to draw lines using the current character color. Choose "flat" to draw lines using the "outer flat border" color. Choose "beveled" to draw 3D lines using the "outer highlight" and "outer shadow" colors.

See also:

[Palette Settings](#)

1.6.4.1.1.1 Attribute Colors

The mapping of visual attributes to colors is displayed as a table with an entry for each possible attribute combination. Select an item from the attribute table to modify or view its properties. Each item has a **background color**, **text color** and **border style**. Sample text drawn using the selected colors and border style is shown in the attribute table. The border style is only available if **text border size** is not "none".

Text border size

The **text border size** is used by "Visual Styles" and determines the thickness of the border drawn around blocks of text. Choices are "thin" (1 pixel) or "thick" (2 pixels). The border is usually drawn *inside* the character cell bounds. Sometimes, especially for wide characters, the pixels used to draw the border overlap the pixels used to draw the character. If this is the case, you can select "thin (ext)" or "thick (ext)", which draws the vertical lines used for the border in the character cell immediately to the left or right of the block. If you use this setting, be sure that your application does not use the characters to the left & right of blocks, otherwise the border lines might overlap those characters.

To select a **border style** for any attribute, the **text border size** must not be "none".

Line draw style

Most of the terminals that AccuTerm emulates support some form of line drawing. AccuTerm supports this function and lets you choose the style of line to use when drawing lines: normal, beveled and flat.

Attribute effects

Since AccuTerm assigns colors to the visual attributes, some of the visual effects are not as important as they were on a real terminal. You can disable the **underline** effect or the **blinking** effect if you like. The assigned colors will still be used for those attributes, but the actual underline or blink will be disabled.

See also:

[Palette Settings](#)

1.6.4.1.1.2 Palette Settings

AccuTerm uses a palette of 16 colors for the background and foreground (text) color of all characters on the screen. There are also 6 colors used for drawing borders when using Visual Styles.

The drop-down list for each of the palette entries displays a list of the 16 standard VGA colors, the system colors (from Windows Display Properties), and a final item called "Custom Color". Click on any item in the list to select that item, or click on Custom Color to open a color selection dialog where you can choose any shade you like for the palette entry.

Since it is possible for the application to request specific foreground and background colors, it is recommended that the color chosen for a particular palette entry is close to the color name associated with that entry.

Note: selecting a theme on the [Colors & Themes](#) panel will replace all of the palette entries with colors compatible with the selected theme.

1.6.4.1.2 Background Settings

AccuTerm has the ability to display a graphic image as the background (or wallpaper) for the terminal screen. If you enable the **use picture for screen background** option, you can enter (or browse for) the name of a file to display as the screen background. A preview of the screen with the selected background picture shows what the selected picture might look like with text written over the background using the currently selected "normal" colors.

Background color transparency

Because detailed pictures can be distracting and might actually obscure the text displayed on the screen, the background picture's transparency can be adjusted. Setting the background color transparency to 0% uses the picture as the screen background. Setting it to 100% shows the normal background color. Any setting in between blends the picture with the normal background color.

Background picture scaling

The background picture will be scaled to fit the screen window. Three scale modes are available:

Stretch: the picture is stretched (or compressed) in both directions to fit the window size.

Fit: the picture's aspect ratio is maintained. It is stretched (or compressed) in one direction to fit the window. Any space not filled by the picture in the other dimension is left black.

Crop: the picture's aspect ratio is maintained. It is stretched (or compressed) in one direction to fit the window. The other dimension is cropped to the window size.

Note: the background picture is only shown in areas of the screen that are "normal". Characters which use a visual attribute other than "normal" obscure the background picture.

1.6.4.2 Keyboard Settings

The **Keyboard Settings** panel is specify keyboard options and open the keyboard programming window.

Locking programmed keys

Most terminals provide commands that allow the host to program the function keys. If you would like the host to be able to reset and reprogram the keys, select the **Unlocked – host can reset or reprogram keys** option. If you would like the host to be able reprogram keys, but not reset them to their default values, select the **Locked – host cannot reset keys to defaults** option. If you would like to prevent the host from resetting or reprogramming keys, select the **Locked – host cannot reset or reprogram keys** option.

Map F11 through F20 to CTRL+F1 through CTRL+F10

Some terminals (like Wyse 50 and 60) have more than 12 function keys on their native keyboard. Since the PC keyboard only has 12 function keys, AccuTerm allows you to map the native **F11**

through **F20** keys to PC keyboard CTRL+F1 through CTRL+F10 (and native SHIFT+F11 to SHIFT+F20 to PC CTRL+SHIFT+F1 through CTRL+SHIFT+F10).

Note: for Wyse, the last native key is F16; for the VT terminals, the last native key is F20; for all others, the last native key is F12. In the preceding paragraph, replace F20 with the last native key for the terminal being emulated.

Start with Caps Lock on

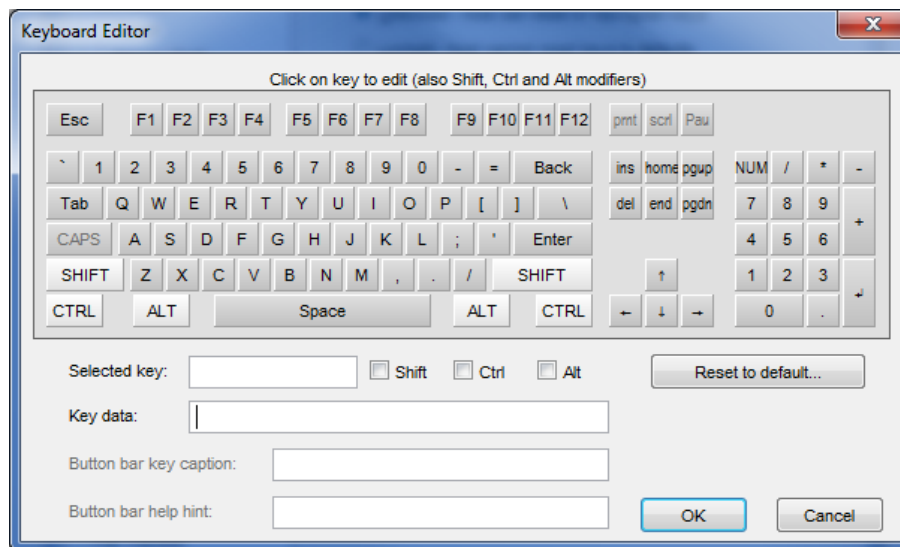
Sometimes applications are designed for upper-case text. If you would like the CAPS LOCK key state to be "on" when the session is opened, check this option.

Program the Keyboard

Click this button to open the [keyboard programming](#) window.

1.6.4.2.1 Keyboard Programming

The **Keyboard Editor** is used to manually program the keyboard. For real terminals, this is usually limited to the function keys F1 to F12. However, AccuTerm supports programming almost any key on the keyboard.



Click on any key to view or edit the programming for that key. The programmed contents of the selected key is copied to the **Key data** field, and the key's name is shown in the **Selected key** field. The state of the **Shift**, **Ctrl** and **Alt** keys reflected by the corresponding check boxes. Clicking the **SHIFT**, **CTRL**, or **ALT** keys on the keyboard editor toggles their state. The new state is reflected in the check boxes.

Key data

If the content of the selected key is the default for that key, the **Key data** field is shown using `normal` characters. If the content is not the default, it is shown in `bold`. Control characters are shown with a caret (^) prefix. You can enter most control characters by simply holding the CTRL key while typing the corresponding alpha key. For example to enter CTRL-M, hold the CTRL key and press the M key. **^M** will be added to the text in **Key data** field.

The carat (^) and backslash (\) characters are used as special prefixes in keyboard programming. If you need to enter either of these characters, just enter two of them (^ or \).

The keys can be programmed with a character sequence which is transmitted to the host when the key is pressed, or a [script command](#), which is executed when the key is pressed. To program a script command, enclose the command in square brackets, for example:

[Beep]

Button bar key caption

One of AccuTerm's features is the ability to show a Function Key Button Bar at the bottom (or top) of the terminal window. The bar contains 12 buttons which can be clicked to perform the same action as the corresponding function key F1 to F12. The caption displayed in the buttons on the Function Key Button Bar can be customized by entering the desired text in this field.

Button bar help hint

Like the **button bar key caption**, each of the buttons can have a custom **help hint** that is displayed when the mouse hovers over the button. Enter the desired help text in this field.

Reset to default...

Clicking the **Reset to default** button opens a **Reset Keys** dialog where you can specify which keys you want to reset: the current key, the normal (A-Z, 0-9, etc.) keys, cursor keys, function keys, function key captions or the entire keyboard. If any option except "current key" is selected, you can restrict the reset to the current key bank. In this case, only the keys with the same Shift, Ctrl and Alt state are reset. Click the **Reset** button to perform the action or **Cancel** to return to the **Keyboard Editor**.

1.6.4.3 Font Settings

Font

The **Font** drop-down list shows all of the available fonts installed on your computer that are suitable for use with AccuTerm. In order for a font to be considered suitable, it must be fixed-pitch (most Windows fonts are not fixed-pitch), and it must support the ANSI character set. AccuTerm uses the open-source **DejaVu Sans Mono** font by default.

The **AccuTerm Legacy** and **AccuTerm Legacy (bitmap)** fonts are also available in case you prefer those instead. Due to limitations of bitmap fonts in Windows, the **AccuTerm Legacy (bitmap)** font is limited to the Latin-1 character set (ISO 8859-1) and does not contain all of the symbol characters.

Size

The **Size** list shows the font sizes available for the selected font. Choose "auto" if you want the font to scale to fit the screen size. This is the default. *Note: when using [Programmers Mode](#) (variable columns and rows), you must select a fixed font size.*

Style

Select either "normal" or "bold" font style.

Keep legible

If this option is checked, and the **Size** is set to "auto", AccuTerm will never scale a font smaller than 5 pixels wide by 7 pixels high.

Character set

Select the character set encoding that is used by the application running on the host. Most terminals

have one or more native character sets - these are always included in the character set encoding list and are the most common choice. The current Windows ANSI character set is also included in the character set list. Select this encoding if your host application is using the same ANSI character set that Windows is using. If your application uses Unicode sends and receives data using UTF-8, select the Unicode (UTF-8) encoding. Other single-byte character sets that are installed in Windows may be also be selected.

For VT220, 320 & 420 emulations, the selected character set is used when AccuTerm receives the SCS (select character set) command to select the UPSS (user-preferred supplemental set).

You should select a font that supports all the characters used by your application. Windows includes a feature known as "Font Linking" which enables the use of characters (actually, glyphs) from a similar font if a required glyph is missing from the selected font. This sometimes yields poor results, so selection of an appropriate font is important. Font Linking is not supported for bitmap fonts.

Euro character

If your host application needs to display the Euro currency symbol, enter the ASCII value of the Euro symbol, as used on your system. *Note: for most character sets there is no standard value for this symbol. Some possible values are 128 (Windows ANSI), 164 (Latin-1 currency and Latin-9 Euro) and 213 (IBM modified code page 850).* Unicode (UTF-8) defines the Euro as U+20AC. If you are using Unicode encoding, the Euro character value is ignored.

1.6.4.4 Sound Settings

Terminal bell

Like the margin bell of antique mechanical typewriters, terminals have the ability to "ring a bell", usually to get the attention of the operator. The ASCII *BEL* control code is universally used to "beep" the terminal. AccuTerm provides great versatility in customizing the sound generated by this command.

Enable terminal bell sound

By default, the "bell" sound is enabled. If your environment requires a silent terminal, you can disable the sound by un-checking this box.

Play Windows sound

By default, AccuTerm will play the wave file assigned to Windows "Default Beep" program event defined in the Sounds applet in Control Panel. If you would like to play the sound defined for a different program event you can select the event from the drop-down list. Click the **Test** button to play the sound immediately.

Play media file


If you would like choose your own WAVE or MP3 file to use as the "bell" sound, select this option. Click the **Browse...** button to browse for the desired file.

Beep using PC speaker

To use a real "beep" as the "bell" sound, select this option and enter the frequency of the beep in Hz, and the duration in milliseconds. The default is 600Hz for 250ms. The frequency must be between 200Hz and 4000Hz. The duration must be between 100ms and 5000ms.

1.6.5 Printer Settings

AccuTerm 8 uses the printer in two ways: First, AccuTerm can print the contents of the terminal screen (Screen Print). The user can perform a Screen Print by using **File ► Print Screen** from the main menu,

by clicking the  button on the toolbar or by using the keyboard shortcut ALT+G. The application on the host can also send a command to AccuTerm to print the screen.

The other way AccuTerm uses the printer is as a "slave printer". Most of the terminals that AccuTerm emulates allowed the user to attach a printer to a special port on the terminal. This printer was known as a "slave printer", which was accessed by the application program differently than a "system printer". The port that the slave printer was attached to was often called an "auxiliary port", so "slave printer" is synonymous with "auxiliary printer" or "aux printer".

AccuTerm can direct data that it receives from the host to the Slave Printer either by receiving the appropriate command from the application or by using **File ► Printer On** from the main menu. The ALT+P keyboard shortcut can also be used to toggle the state of AccuTerm's Slave Printer. A panel in AccuTerm's status bar indicates the current state of the Slave Printer.

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 slave printing in graphics mode. You can select the paper size and orientation used for screen print in graphics mode.

AccuTerm can also use special [print adapter](#) scripts to perform custom processing of slave or screen print jobs.

AccuTerm 8 has separate printer configurations for screen and slave printing. Both configurations contain similar settings which are described in this topic.

Enable printing

Enable or disable slave or screen printing by checking or un-checking this box.

Printer selection

Select the printer to use for slave or screen printing from the **Printer** drop-down list. The first item in the list is always the current default Windows printer. Select this item to use the Windows default printer for slave or screen printing. Even though the name of the current default printer is shown here, selecting this option will track changes to the default printer. If a specific printer is desired, select it from the drop-down list.

Text mode

Select text mode if you want to send raw text to the printer. For slave printing, the print data can contain control codes and escape sequences to control the printer. For screen printing, the screen text is sent to the printer with **CR LF** after each line.

Eject page before / after (text mode only)

Selecting either of these options causes a page eject before or after each print operation. These options

are only available in text mode. Graphics mode always ejects a page after the print operation.

Graphics mode

Graphics mode is normally used for screen printing since the printed output closely resembles the actual screen, including the font used on the screen. Graphics mode may be necessary for slave printing if the selected printer cannot use text mode. Many printers now only support graphics mode.

Page Setup (graphics mode only)

Click this button to select the font, paper size and orientation used for screen or slave printing in graphics mode.

Print background (screen print only)

Select this option to print the screen background when printing the screen in graphics mode.

Print in color (screen print only)

Select this option to print the screen in color. If the **Print background** option is also checked, then the screen background is also printed in color.

Use print adapter script

AccuTerm can utilize *print adapter* scripts to process screen or slave print jobs. The *print adapter* script can launch command-line print utilities such as **Print Wizard** (from Rasmussen Software) to process the print job (file). To use a *print adapter*, select this option and choose the desired adapter script from the drop-down list.

You can create your own custom adapter scripts. Use the supplied PrintWizardPCL.atssc script as a template, and customize your script to process the print job to suite your own requirements.

Print Wizard

Several Print Wizard adapter scripts are supplied with AccuTerm 8. If you have Print Wizard installed on your PC, you can select one of the Print Wizard adapter scripts for screen or slave printing.

Print Wizard PCL Converter: this adapter script can print jobs that contain Hewlett Packard PCL on most any Windows printer, even if the printer does not have native PCL support. This is quite useful if you use PCL-formatted reports and forms in your application, but have newer printers without PCL (or even text) support.

Print Wizard Interactive: this adapter script will open the Print Wizard user interface where the operator can choose the disposition of the print job - save as PDF, email, fax or print.

Print Wizard Kiosk: this adapter script is similar to the Interactive script. The Print Wizard kiosk mode presents more options than the Interactive mode.

Print Wizard PDF Creator: this adapter script creates a PDF file from slave print or screen print jobs. When the print job is closed, the standard Windows "Save as" dialog is displayed to enter the file name for the PDF file.

Encoding for temporary print files

When using a *print adapter*, AccuTerm stores the print job in a temporary file. When the job is closed, the temporary file is passed to the adapter script for processing. You can choose the character set encoding to use for this temporary file. Choose **No encoding** to store the data exactly as received from the host. Other selections convert from the host character set to the selected encoding (ANSI, Unicode

or UTF-8).

Printer job close timeout (slave printer only)

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 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 slave 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 slave port on and off, you may need to adjust this number until things work correctly.

1.6.6 Clipboard Settings

Paste Options: end-of-line

When the clipboard is "pasted" to an AccuTerm session, AccuTerm transmits the clipboard text to the host computer. The **end of line** options determine what AccuTerm does at the end of each line: send **CR** (default), send **LF**, send **CF+LF**, send **TAB**, do nothing, or send 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 sent at the end of each line.

If you do not want the end-of-line character to be sent after the last (or only) line pasted, check the **Do not send line-end after last (or only) line pasted** option.

Paste Options: end-of-text

After pasting the clipboard to an AccuTerm session, AccuTerm can send an optional **end of text** character. Normally, AccuTerm does not transmit any character at this time, but you can select **EOF** (which is the **SUB** or **CTRL+Z** control code), or send 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 sent at the end of a paste operation.

Paste Options: Interline delay

After each line from the clipboard is sent to the host computer, AccuTerm will wait for the specified delay time before sending the next line. The delay time is specified in milliseconds. *Note: this is the same delay setting as the ASCII file transfer.*

Paste Options: Convert curly quotes to straight quotes

Check this option if you would like AccuTerm to automatically convert "curly quotes" into "straight quotes" when pasting. Uncheck this option if you need to paste "curly quotes" into AccuTerm without modification. *Note: "curly quotes" are only supported if your host character set encoding is "Unicode / UTF-8". Otherwise AccuTerm translated them (and any other unsupported character) into an underscore (_) charcter.*

Copy Options: Clear selection after copy

Normally, if there is a selection rectangle shown on the AccuTerm screen, copying the selection to the clipboard leaves the selection intact after copying. Check this option if you would like to automatically

remove the selection from the screen after copying it to the clipboard.

Copy / Paste shortcut

AccuTerm supports standard Copy and Paste functions using the **Edit** menu or the context (right-click) popup menu. You can also choose shortcut keys for these functions: **CTRL+INS / SHIFT+INS** or **CTRL+C / CTRL+V**. However, if you select the shortcut you will no longer be able to use the normal function of the shortcut keys. For example, if you select **CTRL+C / CTRL+V** for your Copy / Paste shortcut, you will not be able to send the **CTRL+C** control character anymore.

1.6.7 Mouse Settings

AccuTerm supports extended mouse functions which can be used to add mouse functionality to your character-based applications. Two mouse extensions are provided: *host mouse reporting*, and *mouse pattern matching*.

Allow mouse input for supported applications

This option determines whether *host mouse reporting* is enabled. When this option is enabled (the default), AccuTerm will enter *mouse input* mode when the host sends the *enable mouse input* command.

Mouse pattern file name

To use *mouse pattern matching*, enter the name of the desired mouse pattern file here, or click **Browse...** to browse for the pattern file. *Note: several sample mouse pattern files are included with AccuTerm in the Samples folder under the main AccuTerm program folder. These files have a .mou file extension.*

See also:

[Mouse support](#)

1.6.8 File Transfer Settings

Use the File Transfer Settings panel to modify how AccuTerm's file transfer functions work. There are sections for ASCII, Kermit and X/Y/Zmodem. It is generally not a good idea to change these settings from their default values unless you have a good understanding of the protocol and the companion file transfer program on the host computer.

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 host computer's ASCII upload program by checking the appropriate option.

ASCII is a "blind" protocol. The host computer cannot tell AccuTerm if it is ready to receive the next line. In order to perform reliable ASCII uploads, you may need to specify an **Interline delay** time. After each line is sent to the host computer, AccuTerm will wait for the specified delay time before sending the next line. The delay time is specified in milliseconds.

*Note: the **Interline delay** setting also applies to the **Edit ► Paste** and **Edit ► Paste From...** commands.*

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 defaults

When AccuTerm's Kermit is negotiating with the host's 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.

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.

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".

Defaults

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. For downloaded text files, select the desired encoding of the downloaded file. Select "no encoding" if no character set translation is desired. Select ANSI, Unicode or UTF-8 if you want to translate downloaded files from the host character set to the selected encoding. Also, select whether to overwrite existing files when downloading. You can change any of these options in the Upload or Download dialog before performing the transfer.

1.6.9 TCL Rules

Several of AccuTerm's MultiValue tools (Upload Wizard, Download Wizard, wED Editor and GUI Designer) require a special MultiValue server utility to function. The server utility is typically initiated from the command line (TCL, ECL, etc.), and settings in this panel define how AccuTerm can verify that the session is "at TCL".

Prompt characters

If TCL can be recognized by one or more particular prompt characters, check the **Prompt characters** checkbox and enter these characters in this field. For example, many MultiValue systems use a colon (:) as the TCL prompt. Some systems change the prompt if a select list is active, for example, to a greater-than-sign (>). In this example, to require the current prompt be either a colon or greater-than-sign for AccuTerm to issue a TCL command, enter ":" and click Add, then enter ">" and click Add.

Cursor row is and **Cursor column is**

If TCL can be recognized by the cursor position, check the **Cursor row is** and/or **Cursor column is** checkbox, then enter the position (or range of positions) where the cursor must be before AccuTerm will issue a TCL command.

TCL check (command, response and timeout)

When the correct prompt character and cursor position are found (as specified in the previous options), AccuTerm will attempt send the command shown in this field to the host to "run at TCL". AccuTerm waits for the specified response to be returned from the host. If the correct response is returned within the timeout period, AccuTerm assumes that the session is at TCL and AccuTerm can run other TCL commands. The default TCL check program is "FTTCL", which is installed with the AccuTerm host programs. When you run FTTCL, it displays the text "***TCL**" on the screen so AccuTerm can verify that the session is "at TCL".

Server command

Once AccuTerm recognizes that the session is at TCL, it sends the **Server command** to the host to initiate "server mode". This mode is required for various AccuTerm MultiValue tools to read and write data from the host. The default server program is "FTSERVER" which is installed with the AccuTerm host programs.

See also:

[Installing the host programs](#)

1.6.10 Window Settings

The **Window Settings** pane is divided into two sections: **Session Window** and **Main Window**. The **Main Window** section is shown only when the **Show advanced options** check box is checked. This section is only available to users with administrator privileges, and only when AccuTerm has been started using "run as administrator".

Session Window


Session title format

When AccuTerm displays the session window title, it may add a copy number or session ID to the title text. The copy number is only added when more than one instance of a particular session file is open. You can select various formats for the title from this drop-down list.

Session title

Normally, AccuTerm uses the session file name as the session window title. If you would like to enter your own title text, enter it in this field.

Session icon

AccuTerm normally shows the standard AccuTerm 8 session icon  in the session tab. If you would like to display a different icon, click the **Change...** button and select the desired icon.

Main Window

Main window title

It is possible to customize AccuTerm's main window title by entering the desired text in this field.

Main window icon



AccuTerm normally shows the standard AccuTerm 8 program icon in the main window title bar. The same icon is used in the Windows task bar and in the ALT+TAB window. If you would like to display a different icon, click the **Change...** button and select the desired icon.

Splash screen

Like many other Windows applications, when AccuTerm starts up, it briefly shows a "splash screen" which contains a graphic image. The splash screen is shown while AccuTerm opens files and performs other preparatory tasks. You can disable this behavior by un-checking the Show splash screen option. You can also select a custom graphic to be used in the splash screen. If you select a custom splash graphic image, its dimensions should be approximately 500x300 pixels.

1.7 Customizing the Menu and Toolbar

A custom AccuTerm 8 menu can be created for the current user (user's personal custom menu) or for all AccuTerm users (master menu). When the **Menu Designer** starts, you can select whether to create (or modify) the current user's menu or, if you have administrator privileges, the master menu. When creating or modifying the master menu, you can force the master menu to override any personal menus by checking the **override** box in the [Menu Properties](#) pane. Otherwise, a user's personal custom menu has priority over the master menu.

Menu Structure

The AccuTerm menu is a hierarchical structure of *Bands* and *Tools*. A band is a collection of related tools organized in a logical manner. There are different types of bands (menu, toolbar, status bar), as well as different types of tools (menu item, toggle item, panel, etc). Bands contain tools which perform actions.

Menu Designer

Use the **Menu Designer** to customize AccuTerm's main menu, popup menu, toolbar and status bar. 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 bar fields.

To start the **Menu Designer**, use **Tools ► Customize Menu** from the main menu.

The **Menu Designer** displays the menu structure using a tree in the left pane of the designer window and properties in the right pane. The root node in the tree is **AccuTerm Menu & Toolbar**. Under the root are nodes for each band in the menu structure. Each menu, toolbar, status bar and session bar is considered a band. There can be only one **Main Menu** band and only one **Session Bar** band. It is possible to have multiple **Toolbar**, **Status Bar**, and **Context Menu** bands, although having multiples may be confusing to the user.

Selecting a node (band or tool) in the menu tree structure causes the right pane of the designer window to display the properties of that node. You customize the selected object by adjusting its properties.

[Menu properties](#)

[Band properties](#)

[Tool properties](#)

New

The **New** button creates a new *band* if the current selection is the root. The **New** button creates a new *tool* if the current selection is a *band* or *Sub-Menu tool* or *Dropdown Button tool*. Clicking the **New** button displays a drop-down menu that allows you to choose the type of item to create.

Delete

The **Delete** button deletes the currently selected item.

Default

The **Default** button resets the entire menu structure back to its default state. Any customizations are lost. You will receive a warning to confirm this action if you click this button.

Cancel

Click the **Cancel** button to close the menu designer and abandon any modifications you have made.

OK

Click the **OK** button to close the menu designer and accept all modifications you have made.

1.7.1 Menu Properties

When the root node (**AccuTerm Menu & Toolbar**) is selected in the menu tree, the properties pane displays the actual menu file name. If the master menu was selected from the designer startup page, the **override** status of the master menu is also displayed in the properties pane.

Master menu override

Selecting the **Master menu override** option causes any existing personal custom menus to be ignored, and prevents the user from creating or modifying his personal menu. If this option is not selected, the master menu is used unless the user has a personal menu. Users will be able to create and modify their personal menu.

1.7.2 Band Properties

Name

Specify the name of the band. The following band names are recognized by AccuTerm and should not be changed: Menu, StatusBar, Toolbar, ContextMenu, SessionBar and TabCtx.

Type

Displays the band type: Main Menu, Popup Menu, Toolbar, Status Bar or Session Bar. You cannot change the band type; the type is determined when you create a new band using the **New** button.

Style

Choose the band style. Usually this is "Normal", but you can choose one of the other styles if you want to display icons in the Session Bar and Status Bar, or display captions in a Toolbar.

Title

The band **Title** is only used if a Toolbar band is "floating" - that is - not "docked" to an edge of the main window. If you change the **Dock position** of a Toolbar to "floating", then this title is displayed at the top of the floating Toolbar window.

Dock position

You can change the dock position of Toolbar and Status Bar bands. Normally, Toolbar bands are docked at the top of the window, and Status Bar bands are docked at the bottom.

Enabled

When a band is Enabled, the user can click on tools in that band. If it is disabled, tools on the band do not respond to user clicks or keyboard shortcuts.

Visible

If you want the user to see the band, set the Visible property. If you clear this property, the band will be hidden. The Visible state of most bands is updated by AccuTerm based on preferences or menu selections.

Large icons

The Large icons property is set at runtime by AccuTerm. Do not alter this property here.

Auto wrap

If you check the Auto wrap property for a Toolbar band, and the window is not wide enough to display all of the Toolbar buttons, the Toolbar will "wrap" at the right edge of the window and show on two or more lines.

See also:

[Tool Properties](#)

1.7.3 Tool Properties

ID

The ID property identifies each tool. Many IDs are built-in and can be selected from the dropdown list. When you create a new tool that does not duplicate the function of a built-in tool, you must specify a unique ID in the range of 10000 to 19999.

Type

The tool Type depends on the type of band the tool is on:

Menu or Popup Menu

Menu Item: when the user clicks a Menu Item, an action is performed.

Sub-Menu: when the user clicks a Sub-Menu, another menu level is displayed.

Toggle Item: when the user clicks a Toggle Item, the item changes state (checked vs. unchecked). When a Toggle Item is "checked", it may display a different icon, or a checkmark if no icon has been designated. When a Toggle Item changes state, its action is performed.

Group Item: a Group Item is like a Toggle Item, except that only one item in the group can be "checked" at a time. A group is delimited by a menu item that has the **Begin group** option checked.

Toolbar

Normal Button: when the user clicks a Normal Button, an action is performed.

Dropdown Button: when the user clicks a Dropdown Button, another popup menu (sub-menu) is displayed.

Toggle Button: when the user clicks a Toggle Button, the button changes state (depressed vs. raised). When a Toggle Button is "checked", it may display a different icon from the "unchecked" state. When a Toggle Button changes state, its action is performed.

Group Button: a Group Item is like a Toggle Button, except that only one button in the group can be "checked" at a time. A group is delimited by a button item that has the **Begin group** option checked.

Status Bar

Normal panel: displays text and/or icon in the Status Bar.

Num Lock panel: displays the state of the Num Lock key in the Status Bar.

Caps Lock panel: displays the state of the Caps Lock key in the Status Bar.

Scrl Lock panel: displays the state of the Scroll Lock key in the Status Bar.

Time panel: displays the current time in the Status Bar.

Date panel: displays the current date in the Status Bar.

Date/Time panel: displays the current time and date in the Status Bar.

Session Bar

A Session Bar does not have any design-time tools. The Session Bar buttons are created automatically.

Style

The tool Style determines whether text, icon or both are displayed. You can also choose "normal", which will display the most common style for the tool and band type.

Caption

The tool Caption text is displayed in Menu and Status Bar tools. It can optionally be displayed for a Toolbar button. For Menu tools, you can create an "accelerator" for the item by inserting an ampersand (&) before the letter you want to use for the accelerator in the Caption.

Alignment

For Status Bar tools, choose the text alignment: Left, Right or Center.

Border style

For Status Bar tools, choose the panel border style: Inset, Raised or None.

Panel size

For Status Bar tools, choose the panel size: Fixed Width, Fit Contents or Spring. If you choose Fixed Width, and set the Minimum Panel Width to zero, the panel will be sized using the text in the Caption property. If you choose Fit Contents, the panel will adjust size any time the Caption text is updated. If you choose Spring, the panel size is like Fit Contents, but varies with window size. All Spring panels are resized when the window size changes.

Minimum panel width

Specify the size of a Fixed Width panel, or the minimum size for a Fit Contents or Spring panel. This value is in pixels.

Shortcut

For Main Menu tools only, you can select a keyboard shortcut for that tool. This shortcut is usually combined with Ctrl, Shift and Alt modifier keys. Note that the shortcut overrides any other use of the key. For example, if you designate Ctrl+C as a shortcut, AccuTerm will be unable to send the CTRL+C control code when you press CTRL+C.

Tool tip

Tool tip text is displayed when you leave the mouse over a Toolbar tool or Status Bar panel.

Action

The Action field determines what happens when the user clicks the tool. For built-in tools, the Action is the same as the ID. For user-defined tools, you can either enter a string to send to the host, or a script, in this field. The syntax is exactly the same as [function key programming](#).

Enabled

When a tool is Enabled, the user can click on it. If it is disabled, the click is ignored. For built-in tools,

AccuTerm will adjust this property depending on the program and session state.

Visible

If you want the user to see the tool, set the Visible property. If you clear this property, the tool will be hidden. For built-in tools, AccuTerm will adjust this property depending on the program and session state.

Checked

This is the initial state of a Toggle Item, Group Item, Toggle Button or Group Button. For built-in tools, AccuTerm will adjust this property depending on the program and session state.

Begin group

If the Begin group property is set, there will be a separator before the menu item or toolbar button. This property is also used to identify which buttons belong to a group of Group Items or Group Buttons.

Select Image...

Click the Select Image button to open the [Image Selection](#) dialog to choose an icon for a button or menu item.

See also:

[Band Properties](#)

1.7.4 Image Selection

When you click the **Select Image...** button in the **Tool properties** panel, the **Image Selection** dialog is displayed. Use this dialog to select an icon to display for the tool. You can select both normal (un-checked) and checked icons. For each tool, the image can be one of the built-in icons, or you can specify a file name for the image.

Image source

Choose the image source, or specify that the tool does not have an icon. AccuTerm 8 uses an "icon font" for its built-in tools. You can copy/paste the image file name from one of the built-in tools to use for your own custom tool. You can also choose your own font icon, provided you have the icon font installed in Windows. Use the same format as the built-in tools (Font Name # Unicode Character Code in Hex).

Normal image

Check this box if the tool has a normal (un-checked) icon. Enter the image file name (if the tool's icon is loaded from a file at runtime), or the image ID if the tool uses a built-in image. Click the **Browse** button to browse for image files or built-in images.

Checked image

Check this box if the tool has a checked icon that is displayed when the tool is in its "checked" state. Enter the image file name (if the tool's icon is loaded from a file at runtime), or the image ID if the tool uses a built-in image. Click the **Browse** button to browse for image files or built-in images.

1.8 File Transfer

AccuTerm has the ability to transfer files between your computer and the host computer using one of the five built in file transfer protocols: ASCII, Kermit, Xmodem, Ymodem and Zmodem. The manual (non-MultiValue) file transfer features are accessed from the **Tools ► Upload** or **Tools ► Download** from the main menu.

AccuTerm also includes a library of file transfer utility programs for use with MultiValue based host systems. For users of MultiValue based systems, using these utilities is the recommended method of exchanging data between your PC and the host system. If you have the MultiValue host programs installed, you can use the File Transfer Wizard to guide you through the file transfer process.

See also:

[Manual Upload](#)

[Manual Download](#)

MultiValue File Transfer Wizard

[MultiValue File Transfer Programs \(installation and use\)](#)

1.8.1 File Download

Download transfers a file (or files) from the host computer (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, use **Tools ► Download** from the main menu to open the **Download** dialog, 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** dialog.

Note: if you are using a MultiValue-based host system (jBASE, QM, D3, Pick, UniVerse, Unidata, etc.), AccuTerm includes a library of MV/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 [MultiValue File Transfer and Server Programs](#) for details on installing and using these programs.

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.

Encoding

When the transfer mode is text, select the encoding (character set) to use for the downloaded file. Select None, ANSI, Unicode or UTF-8.

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.*

1.8.2 File Upload

Upload transfers a file (or files) from the workstation to a remote machine (host). 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, use **Tools ► Upload** from the main menu to open the **Upload** dialog, select the appropriate protocol, select the file(s) to upload, then press ALT+L or click the **Upload** button.

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

Note: if you are using a MultiValue-based host computer (D3, Pick, UniVerse, Unidata, QM, jBase, etc.), AccuTerm includes a library of MV/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 [MultiValue File Transfer and Server Programs](#) for details on installing and using these programs.

File Selection

Click the **Browse** button to select the file(s) to upload. If you have selected a protocol which can upload more than one file (**Kermit**, **Ymodem**, or **Zmodem**), 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: **ASCII**, **Kermit**, **Xmodem**, **Ymodem** or **Zmodem**.

The **ASCII** 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 **ASCII** 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 **CR**, **LF** or **CR LF** by selecting the desired option in the [File Transfer](#) category of the **Settings** dialog box (**Tools ► Settings** 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 **Kermit** protocol provides error detection and correction. **Kermit** is a "batch" protocol, so you can send multiple files. **Kermit** can send either text or binary files; text files are converted to the format required on the receiving machine. **Kermit** 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 **File Transfer** category of the **Settings** 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.

When **text** mode is selected, AccuTerm can **translate** the file to the host character set as the file is uploaded. When this option is selected, AccuTerm examines the first few bytes of the file to determine its encoding (ANSI, Unicode or UTF-8), and converts the content to the host encoding automatically.

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.*

1.8.3 Import & Export MultiValue Data

To make the most effective use of AccuTerm with your MultiValue based host system, you should install the AccuTerm host programs. These programs implement the Kermit and ASCII file transfer protocols, coordinate the transfer of data between your computer and the host, import and export data in a variety of formats and provide a simple host-to-host transfer facility. See [Installing MultiValue Host Programs](#) for information on installing the host programs.

In order to understand how to best use AccuTerm's MultiValue file transfer features, the differences between the PC and MultiValue file systems must be examined. Traditional PC files may be

programs, documents or data files; in the typical MultiValue system, programs and documents are stored as individual items (records) in a file, whereas a data file is collection of related items (records) in a file, with an associated file dictionary.

When exchanging data between your computer and your MultiValue host, it is important to realize the difference between “document files” and “data files”.

MultiValue File Transfer Wizard

Use the File Transfer Wizard to simplify the process of uploading or downloading data and documents between your PC and MultiValue host. The wizard uses a series of simple questions to guide you through the file transfer process. You can use the wizard to help you upload or download data or document files.

MultiValue File Transfer Utilities

When transferring “document files” between the PC and MultiValue host, use AccuTerm’s **FT** utility. This utility transfers MultiValue *items* to PC *files* and vice versa. The key concept is that each MultiValue *item* becomes a separate *file* on the PC.

When transferring “data files” between your PC and your MultiValue host, use the **FTD** utility. This utility transfers a set of MultiValue *items* to a single PC *file* and vice versa. The key concept is that a set of *items* on the host becomes a single *file* on the PC.

You can also use AccuTerm to transfer data between two MultiValue hosts. The **FTPICK** utility is used to transfer data from one MultiValue machine to another, either directly (using two concurrent AccuTerm sessions), or using an intermediate PC file.

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 file.

MultiValue File Transfer Functions:

[Installing MultiValue Host Programs](#)

[Updating the MultiValue Host Programs](#)

[Downloading documents to your PC](#)

[Uploading documents from your PC](#)

[Downloading data files to your PC \(export\)](#)

[Uploading data files from your PC \(import\)](#)


[Transferring data between two MultiValue hosts](#)

[Integrating file transfer functions into MultiValue applications](#)

1.9 Data Capture

Initiating Data Capture

AccuTerm is capable of capturing data to a file data. The source of the captured data can be either the entire stream of characters received by the workstation (possibly filtering out control characters), or the data which would normally be printed to the slave printer. To start data capture, select **Tools ►**

Capture from the main menu or click the  button on the toolbar. Select the capture file and options in the **Begin Capture** dialog.

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. If you select a file or folder for which you do not have the necessary permissions to update, 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 and escape sequences. 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.

Filter


Select the **Text only (strip control codes)** option if you want control characters and escape sequences filtered out of the captured data.

Encoding

When capturing text data, select the encoding (character set) to use for the capture file. Select None, ANSI, Unicode or UTF-8.

To initiate the data capture, click the **Start Capture** button.

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** dialog. Click the **End Capture** button to terminate the capture operation and close the capture file, or click the **Continue** button to resume capturing.

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

1.10 Troubleshooting

Common problems and solutions are described here. If you cannot find the solution to your problem, you should visit Zumasys' web site and check the FAQs (frequently asked questions) to see if a solution is available. Otherwise, you can send an email to accuterm@zumasys.com.

Unanticipated Errors

If AccuTerm encounters an unanticipated error condition, the "Unanticipated Error" dialog box will be displayed showing the error source and description. You can choose to Ignore the error, Retry the failing operation, or Fail (terminates the failing operation).

When an unanticipated error occurs, a log of the error is written to the main program directory (Atwin80.log). If you have email capabilities, you can email this log file to accuterm@zumasys.com along with a description of the conditions surrounding the error (What you were doing when the error occurred? What version of Windows? Any other programs open?).

Network Problems

Most network problems are caused by incorrect network settings or a network failure. AccuTerm 8 only lets you specify the host name or IP address and optional host port number. Other network settings are beyond the scope of this document.

If you receive a "Timeout" error when attempting to open a network connection, the host computer you are attempting to connect to may not be responding. AccuTerm's default timeout is 15 seconds, which is sufficient for most hosts even across the Internet.

If you receive a "Host unreachable" error then the IP address or subnet mask of your machine may be incorrect. Also, check the IP address of the host you are attempting to connect to.

If your connection drops after an idle period, it may be caused the an "idle timeout" on the host (or a router). If this happens, you may be able to enable the [keepalive](#) option for your network connection.

ReZume

AccuTerm 8 includes a premium feature, ReZume, which allows you to "resume" your session after a temporary network disconnect. See [ReZume](#) for more details.

1.11 Contacting Zumasys

Zumasys customer service department may be reached by phone, fax, email or the web.




Zumasys, Inc.
1050 Calle Amanecer - Suite A
San Clemente, CA 92673
USA

Phone: 1 (866) 986-2797
Fax: 1 (949) 334-0288
Email: accuterm@zumasys.com
Web: www.zumasys.com

2 AccuTerm 8 Automation

2.1 Macro Recorder

Recording a Macro

You can use the **Macro Recorder** to create an AccuTerm macro by recording a sequence of actions, including sending keystrokes to the host and waiting for responses from the host. To start recording a new macro, use **Tools ► Macro ► Record New Macro** from the main menu. When you are finished recording the macro, click the  button on the Macro Recorder toolbar. You can temporarily suspend recording by clicking the  button on the Macro Recorder toolbar. To resume recording, click the  button.

When you are finished recording a macro, the **Save Macro** dialog will be displayed. Here you can give your recorded macro a name (up to 32 characters, letters, numbers or underscore characters only, beginning with a letter), a short description (up to 32 characters) which is displayed in the **Tools ► Macro ► Run Macro** sub-menu, a detailed description and a shortcut key.

The recorded macro is saved in the "AccuTerm Macros.atsc" script file when you click the **Save** button. If you do not have a macro script file open yet, you will be prompted to select the file name for your macro file. When you start AccuTerm, the macro script file (if it exists) is automatically loaded.

Macros are recorded as a sequence of Visual Basic for Applications (VBA) statements.

Running a Macro

To run a macro, select the desired macro from **Tools ► Macro ► Run Macro** menu. Alternatively, if the macro has a shortcut key assigned to it, simply press the shortcut key combination to run the macro.

Editing a Macro

You can edit your recorded macros by selecting **Tools ► Macro ► Edit Macros...** from the main menu. The macro script will be displayed in the macros tab of the **AccuTerm Script Editor** window. Macros are written using Visual Basic for Applications (VBA) language. For more information about VBA scripting, see the [Scripting](#).

2.2 Scripting

AccuTerm 8 includes a powerful scripting language based on 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.

Script Editor

Although the **Script Editor** is not the only way to create and use VBA scripts with AccuTerm, it is a convenient place to learn about AccuTerm's scripting features. To access the **Script Editor**, use **Tools ► Script Editor** from AccuTerm's main menu.

The **Script Editor** window has a standard Windows menu and toolbar with tabs for each of the two script files that AccuTerm supports: Main Script and Macro. User-created script files are loaded into the **Main Script** tab; recorded macros are loaded into the **Macro** tab.

To quickly locate a function or subroutine in the code pane of the script editor, select it from the **Proc** drop-down list. If you would like the **Run** command (or the **Debug ► Step Into**) to begin execution with the selected procedure, check the **Run selected procedure** check-box (normally **Run** begins execution with `sub Main()`).

Creating a Script

Scripts are created by entering VBA statements into the script code window. Scripts usually begin with `sub Main()` and end with `end sub`. Enter any valid 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 three 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 (see [Tool Properties](#)).

Saving a Script File

After the script has been created or modified, save the script by clicking the  button or use **File ► Save** or **File ► Save As** from the Script Editor menu.

Loading a Script File

To load a script file, click the  button use select **File ► Open** from the Script Editor menu.

Loading and Running a Script File

To load and run a script file, use **File ► Run** from the Script Editor menu. Enter or browse for 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 **File ► Close** from the Script Editor menu to close the script window. If there are any unsaved changes to the current script, you will be prompted to save changes.

Instead of closing the Script Editor window, you can hide it by using **Window ► Hide** or activate the main AccuTerm window by using **Window ► AccuTerm 8**.

Printing a Script

Select **File ► Print** from the Script Editor menu to print the current script. Select **File ► Print Setup** to select the printer used to print the script.

Editing Scripts

The Script Editor **Edit** menu provides many editing functions such as **Undo**, **Redo**, **Cut**, **Copy**, **Paste**, **Delete**, **Select all**, **Indent**, **Outdent**, **Find** and **Replace**. The font used in the script window may be changed by using **Edit ► Font**, and the dialog editor may be invoked by using **Edit ► User Dialog**.

See also:

[AccuTerm Object Reference](#)

[Running a Script](#)

[Debugging a Script](#)

[Controlling AccuTerm with Scripts](#)


Script Editor Reference

Script Language Reference


[Script Language Extensions](#)

2.2.1 Running a Script

To run a script from the Script Editor, create a new script or load an existing script into the script

window. Then use **Run ► Start (resume)** from the Script Editor menu or click the  toolbar button or press the F5 key (to single step the script, see [Debugging a Script](#).) To terminate the script, use **Run**

► **End** from the menu or click the  toolbar button. To suspend execution, use **Run ► Pause** from

the menu or click the  button on the toolbar. To resume execution, use **Start ► (resume)** or click

the  button.

Unless the **Run selected procedure** check-box is checked, script execution begins with `sub Main()`.

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 **Properties**). When AccuTerm starts, it loads the script into a hidden script window, and executes its `Main` subroutine. You can use a command line script to open up a number of sessions, the log a user on, etc.

You can the file association for AccuTerm's script files (.atsc) to start AccuTerm from a script file (or a shortcut to a script file). This is essentially the same as running a script from the command line.

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 Programming](#)) with script statements enclosed in brackets []. Multiple statements can be separated by a colon (:). A script stored in a function key can contain a maximum of 250 characters. If a more complex script is to be executed from a function key, create it as a separate script file and use the [Chain](#) statement in the function key to run the script file. For example:

```
[Chain "MyComplexScript.atsc"]
```

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](#), 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`

or

`ESC STX R script CR`

See AccuTerm Programming for details.

2.2.2 Debugging a Script


The Script Editor menu and toolbar provide functions for executing, interrupting, resuming and single stepping a script, and for setting breakpoints and watch expressions.


Run ► Start (resume) (





or F5) begins execution at `Sub Main` or resumes execution at the current


statement. If the **Run selected procedure** box is checked, then the procedure in the **Proc** drop-down list is started instead of `Sub Main`.

Run ► Pause ( or CTRL+BREAK) will interrupt execution and enter Break mode.


Debug ► Step Into ( or F8) will execute the next statement and then enter Break mode. If the next statement calls into a subroutine or function, **Step Into** will Break in the called procedure.

Debug ► Step Over ( or SHIFT+F8) is the same as **Step Into** except that when it reaches a call into a subroutine or function, it will not step into the called procedure. The procedure will run, and Break mode will be entered at the next statement in the current procedure.

Debug ► Step Out ( or CTRL+F8) causes a called procedure to continue running and enter Break mode when the called procedure returns.

Debug ► Toggle Breakpoint ( or F9) sets or clears a breakpoint at the current statement. Lines containing breakpoints are marked with a large dot in the left margin.

Debug ► Add Watch adds a watch expression based on the variable or function at the cursor position. Use **Step Into** to start the script before adding a **Watch** expression, otherwise a warning is displayed. To remove a watch expression, click on the **Watch** tab, select the expression you want to delete, and press the DELETE key.

Debug ► Quick Watch () will show the value of a variable or expression in the **Immediate** window. Position the cursor on the desired expression, then use the menu or toolbar button to view the value.

2.2.3 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 Options** or **Files & Folders** 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 [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 topic.

[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 topic.

`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.

2.2.4 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](#)
- [AppClose statement](#)
- [AppFind function](#)
- [AppGetActive function](#)
- [AppGetPosition statement](#)
- [AppGetState function](#)
- [AppHide statement](#)
- [AppList statement](#)
- [AppMaximize statement](#)
- [AppMinimize statement](#)
- [AppMove statement](#)
- [AppRestore statement](#)
- [AppSetState statement](#)
- [AppShow statement](#)
- [AppSize statement](#)
- [Chain statement](#)
- [Common collection](#)
- [FileExists function](#)
- [Item function](#)
- [ItemCount function](#)
- [Line function](#)
- [LineCount function](#)
- [OpenFileName function](#)
- [Pause statement](#)
- [Random function](#)
- [SaveFileName function](#)
- [Sleep statement](#)
- [Word function](#)
- [WordCount function](#)

2.2.4.1 Extensions

2.2.4.1.1 AppActivate statement

Syntax:

```
AppActivate( title$ )
```

Description:

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.

2.2.4.1.2 AppClose statement

Syntax:

```
AppClose names$
```

Description:

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

2.2.4.1.3 AppFind function

Syntax:

```
AppFind( name$ )
```

Description:

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

2.2.4.1.4 AppGetActive function

Syntax:

```
AppGetActive( )
```

Description:

Returns a String containing the name of the active application.

2.2.4.1.5 AppGetPosition statement

Syntax:

```
AppGetPosition X, Y, width, height [ , name$]
```

Description:

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

2.2.4.1.6 AppGetState function

Syntax:

AppGetState(*name\$*)

Description:

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

2.2.4.1.7 AppHide statement

Syntax:

AppHide *name\$*

Description:

Hides the named application.

2.2.4.1.8 AppList statement

Syntax:

AppList *AppNames\$*()

Description:

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.

2.2.4.1.9 AppMaximize statement

Syntax:

AppMaximize *name\$*

Description:

Maximizes the named application.

2.2.4.1.10 AppMinimize statement

Syntax:

AppMinimize *name\$*

Description:

Minimizes the named application.

2.2.4.1.11 AppMove statement

Syntax:

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

Description:

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

2.2.4.1.12 AppRestore statement

Syntax:

AppRestore *name\$*

Description:

Restores the named application.

2.2.4.1.13 AppSetState statement

Syntax:

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

Description:

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.

2.2.4.1.14 AppShow statement

Syntax:

AppShow *name\$*

Description:

Shows the named application.

2.2.4.1.15 AppSize statement

Syntax:

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

Description:

Sets the size of the named application.

2.2.4.1.16 Chain statement

Syntax:

Chain *filename* [*arguments*]

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

Description:

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:*

Chain "C:\ATWIN\SCRIPTS\MYSCRIPT.SCR FOO IMAGE XX.JPG"
--

2.2.4.1.17 Command() function

Syntax:

Command()

Data type: string

Description:

This function returns the command line. If the script was invoked from another script using the `Chain` statement, the command line is taken from the `Chain` statement argument.

2.2.4.1.18 FileExists function

Syntax:

FileExists(*name\$*)

Description:

Returns True if file exists, otherwise returns False.

2.2.4.1.19 Item function

Syntax:

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

Description:

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 *delimiter\$* parameter.

2.2.4.1.20 ItemCount function

Syntax:

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

Description:

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 *delimiter\$* parameter.

2.2.4.1.21 Line function

Syntax:

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

Description:

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

2.2.4.1.22 LineCount function

Syntax:

LineCount(*text\$*)

Description:

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

2.2.4.1.23 OpenFileName function**Syntax:**

```
OpenFileName( [ title$ [, extension$] ] )
```

Description:

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 file types and extensions - **type:ext[,ext][;type:ext[,ext]]...** where **type** is a description of the type of file, and **ext** is the extension pattern. For example, to display "All Files" and "Documents", the **extension\$** argument might be: "All files:*. *;Documents:*.txt,*.doc"

2.2.4.1.24 Pause statement**Syntax:**

```
Pause Seconds
```

Description:

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.

2.2.4.1.25 Random function**Syntax:**

```
Random( min , max )
```

Description:

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

2.2.4.1.26 SaveFileName function**Syntax:**

```
SaveFileName( [ title$ [, extension$] ] )
```

Description:

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 file types (see [OpenFileName](#) for details).

2.2.4.1.27 Shell function**Syntax:**

```
Shell( command$, [ WindowType ] )
```

Description:

Executes a Windows or DOS command. If **command** specifies a file or URL, the file or URL is opened

using the default application defined for that file. The return value is non-zero if the function succeeds. The **WindowType** argument is optional and specifies the window state of the executed program: 0=hidden, 1=normal, 2=minimized (default), 3=maximized, 4=normal but not activated, 6=minimized but not activated.

2.2.4.1.28 Sleep statement

Syntax:

Sleep *milliseconds*

Description:

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.

2.2.4.1.29 Word function

Syntax:

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

Description:

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

2.2.4.1.30 WordCount function

Syntax:

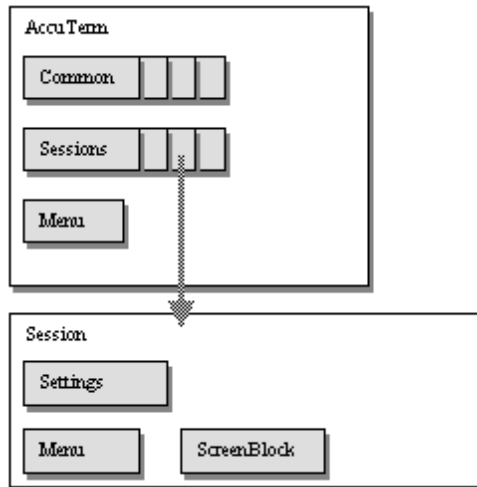
WordCount (*text\$*)

Description:

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

2.3 Object Reference

AccuTerm 8 exposes a rich COM/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

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, 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

A *method* is a subroutine or function call. To reference a method, simply append a period (.) followed by the method name to an object variable of the appropriate type. Function 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 8 includes a type library, `Atwin80.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
```

See also:[AccuTerm object](#)[Menu object](#)[ScreenBlock object](#)[Server object](#)[Session object](#)[Sessions collection](#)[Settings object](#)[AccuTerm constants](#)

2.3.1 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.

AccuTerm properties[ActiveSession property](#)[AutoClose property](#)[Common property](#)[CustomMouseTable property](#)[FuncBarPos property](#)[FuncBarStyle property](#)[FuncBarVisible property](#)[Height property](#)[Left property](#)[Menu property](#)[NoCloseWarning property](#)[NoConnectWizard property](#)[NoWelcomePage property](#)[ProductLicenseType property](#)[ProductName property](#)[ProductRelease property](#)[ProductSerialNumber property](#)[RecentListSize property](#)[RegisteredCompany property](#)[RegisteredLocation property](#)[RegisteredUser property](#)[SessionBarVisible property](#)

[Sessions property](#)
[SingleInstance property](#)
[StatusLineVisible property](#)
[ToolbarVisible property](#)
[Top property](#)
[Visible property](#)
[Width property](#)
[WindowState property](#)

AccuTerm methods

[Activate method](#)
[Arrange method](#)
[Close method](#)
[Hide method](#)
[Move method](#)
[Resize method](#)
[SettingsDialog method](#)
[Show method](#)
[Terminate method](#)

2.3.1.1 AccuTerm properties**2.3.1.1.1 ActiveSession property****Syntax:**

Set *object* = AccuTerm.ActiveSession

Data type: Session object

Description:

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.

2.3.1.1.2 AutoClose property**Syntax:**

`AccuTerm.AutoClose`

Data type: boolean

Description:

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

See also:

[Session AutoClose property](#)

2.3.1.1.3 Common property**Syntax:**

`AccuTerm.Common`

Data type: collection object

Description:

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.

2.3.1.1.3.1 Common collection**Syntax:**

```
Common.Clear  
Common(key) = value  
variable = Common(key)
```

Description:

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.

2.3.1.1.4 CustomMouseTable property**Syntax:**

```
AccuTerm.CustomMouseTable
```

Data type: string

Description:

This is the file name of the global mouse table file. See [Mouse Pattern Matching](#) for more information on using a custom mouse table.

See also:

[Session CustomMouseTable property](#)

2.3.1.1.5 FuncBarPos property**Syntax:**

```
AccuTerm.FuncBarPos
```

Data type: integer

Description:

Set to zero to show the function key bar at the bottom of the window, or 1 to show the bar at the top of the window.

2.3.1.1.6 FuncBarStyle property**Syntax:**

```
AccuTerm.FuncBarStyle
```

Data type: integer

Description:

Set to 1 to display the function key number as a tiny number in the corner of the function key buttons, or 0 to not display the key number.

2.3.1.1.7 FuncBarVisible property

Syntax:

`AccuTerm.FuncBarVisible`

Data type: boolean

Description:

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

2.3.1.1.8 Height property

Syntax:

`AccuTerm.Height`

Data type: long

Description:

This is the application window height in pixels.

2.3.1.1.9 LargeIcons property

Syntax:

`AccuTerm.LargeIcons`

Data type: boolean

Description:

Set to True (non-zero) to show large icons in the toolbar, otherwise small icons are used.

2.3.1.1.10 LayoutFilename property

Syntax:

`AccuTerm.LayoutFilename`

Data type: string

Description:

This is the file name of the currently open layout file, or null if no layout is open. This property is read-only.

2.3.1.1.11 Left property

Syntax:

`AccuTerm.Left`

Data type: long

Description:

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

2.3.1.1.12 Menu property

Syntax:

Set *menu_object* = AccuTerm.Menu

Data type: menu object

Description:

This returns a reference to the application Menu object, which may be used to customize AccuTerm's menus and toolbars.

See also:

[Menu Object](#)

2.3.1.1.13 NoCloseWarning property

Syntax:

AccuTerm.NoCloseWarning

Data type: integer

Description:

Set to True (non-zero) to suppress the disconnect warning when closing a session.

2.3.1.1.14 PhoneBookName property

Syntax:

AccuTerm.PhoneBookName

Data type: string

Description:

This is the file name of the Auto Dialer phone directory.

2.3.1.1.15 ProductLicenseType property

Syntax:

AccuTerm.ProductLicenseType

Data type: integer

Description:

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

2.3.1.1.16 ProductName property

Syntax:

AccuTerm.ProductName

Data type: string

Description:

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

2.3.1.1.17 ProductRelease property

Syntax:

`AccuTerm.ProductRelease`

Data type: string

Description:

This read-only property returns the current release of AccuTerm 8.

2.3.1.1.18 ProductSerialNumber property

Syntax:

`AccuTerm.ProductSerialNumber`

Data type: long

Description:

This read-only property returns the serial number of AccuTerm 8.

2.3.1.1.19 RecentListSize property

Syntax:

`AccuTerm.RecentListSize`

Data type: integer

Description:

Specifies the number of entries in the recent file list.

2.3.1.1.20 RegisteredCompany property

Syntax:

`AccuTerm.RegisteredCompany`

Data type: string

Description:

This read-only property returns the registered company name.

2.3.1.1.21 RegisteredLocation property

Syntax:

`AccuTerm.RegisteredLocation`

Data type: string

Description:

This read-only property returns the registered location.

2.3.1.1.22 RegisteredUser property

Syntax:

`AccuTerm.RegisteredUser`

Data type: string

Description:

This read-only property returns the registered user name.

2.3.1.1.23 SessionBarVisible property

Syntax:

`AccuTerm.SessionBarVisible`

Data type: integer

Description:

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

2.3.1.1.24 Sessions property

Syntax:

`AccuTerm.Sessions`

Data type: collection of session objects

Description:

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

2.3.1.1.25 ShortcutNewWindow property

Syntax:

`AccuTerm.ShortcutNewWindow`

Data type: integer

Description:

Set to True (non-zero) to open shortcuts in a new window, or False (0) to open shortcuts as a new tab in the most-recently-active AccuTerm window.

2.3.1.1.26 SingleInstance property

Syntax:

`AccuTerm.SingleInstance`

Data type: integer

Description:

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

2.3.1.1.27 StatusLineVisible property

Syntax:

`AccuTerm.StatusLineVisible`

Data type: integer

Description:

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

2.3.1.1.28 ToolbarVisible property

Syntax:

`AccuTerm.ToolbarVisible`

Data type: integer

Description:

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

2.3.1.1.29 Top property

Syntax:

`AccuTerm.Top`

Data type: long

Description:

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

2.3.1.1.30 TrackKeyboardState property

Syntax:

`AccuTerm.TrackKeyboardState`

Data type: integer

Description:

If this property is True, AccuTerm tracks the state of the CapsLock and ScrollLock keys. When tracking is enabled, the state of these keys is saved and restored when the user switches to and from the AccuTerm application.

2.3.1.1.31 Visible property

Syntax:

AccuTerm.Visible

Data type: boolean

Description:

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

2.3.1.1.32 Width property

Syntax:

AccuTerm.Width

Data type: long

Description:

This is the application window width in pixels.

2.3.1.1.33 WindowState property

Syntax:

AccuTerm.WindowState

Data type: integer

Description:

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

See also:

[Session WindowState property](#)

2.3.1.2 AccuTerm methods

2.3.1.2.1 Activate method

Syntax:

AccuTerm.Activate

Description:

Makes AccuTerm the active application.

See also:

[Session Activate method](#)

2.3.1.2.2 Arrange method

Syntax:

AccuTerm.Arrange [*style*]

Description:

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

2.3.1.2.3 Close method**Syntax:**

AccuTerm.Close [**prompt**]

Description:

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

See also:

[Terminate method](#)

[Session Close method](#)

2.3.1.2.4 Hide method**Syntax:**

AccuTerm.Hide

Description:

This method makes AccuTerm's main window invisible.

See also:

[Session Hide method](#)

2.3.1.2.5 Move method**Syntax:**

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

Description:

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

See also:

[Session Move method](#)

2.3.1.2.6 OpenLayout method**Syntax:**

AccuTerm.OpenLayout **filename**

Description:

This method opens the specified layout file. When using this method to open a layout file, any script attached to the layout file is ignored.

2.3.1.2.7 Resize method

Syntax:

AccuTerm.Resize *width, height*

Description:

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

See also:

[Session Resize method](#)

2.3.1.2.8 SaveLayout method

Syntax:

AccuTerm.SaveLayout [*filename*]

Description:

This method saves the current layout in the specified layout file. If *filename* is not specified, the currently open layout file is used. If there is no currently open layout file, the user is prompted for a file name.

2.3.1.2.9 SettingsDialog method

Syntax:

AccuTerm.SettingsDialog [*InitSel*, [*Categories*]]

Description:

This method displays AccuTerm's **Settings** dialog box. You can specify the initially selected item *InitSel* (see table below), and a list of **Categories** to show in the categories tree. The **Categories** list is a string with a single letter for each main category to display. Letters which may be used for *InitSel* or **Categories** are:

Category code	Tab code (2K2)	Category
O	G	General Options
A	G	Files & Folders
D	D	Connection
D1		Serial
D2		PicLan
D3		Telnet
D4		Modem
D5		SSH
D7		SSL
T	T	Term Type
S	S	Screen
C	C	Colors & Palette
C0		Color Theme
C1		Attribute Colors
C2		Color Palette
B	C	Background
K	K	Keyboard

Category code	Tab code (2K2)	Category
F	F	Font & Character Set
P	P	Printer
P1		Screen Printer
P2		Slave Printer
Q	M	Clipboard
H	M	Mouse
E	M	Sound
X	X	File Transfer
W	G	Windows
R	R	TCL Rules

For compatibility with AccuTerm 2K2, you can use the alternative **Tab** code in place of the new **Category** code, as shown in the table above. If you do not specify a value for *InitSel*, the previously active category is initially selected. If you do not specify a value for *Categories*, all available categories are displayed.

2.3.1.2.10 Show method

Syntax:

`AccuTerm.Show`

Description:

Makes AccuTerm's main window visible.

See also:

[Session Show method](#)

2.3.1.2.11 Terminate method

Syntax:

`AccuTerm.Terminate` [*prompt*]

Description:

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

See also:

[Close method](#)

[Session Close method](#)

2.3.2 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 [MnuTool](#) objects.

Menu properties

[Count property](#)

[Item property](#)

[MnuBands property](#)

See also:

[AccuTerm Menu property](#)

[Session Menu property](#)

2.3.2.1 Menu properties

2.3.2.1.1 Count property

Syntax:

menu.Count

Data type: integer

Description:

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

2.3.2.1.2 Item property

Syntax:

*menu.Item(**toolID**)*

Data type: `MnuTool` object

Description:

This array property returns a reference to a `MnuTool` object for the specified **toolID**. Each menu item, toolbar button, and status bar field has a **toolID** that identifies the tool (see [Customizing the Menu and Toolbar](#) for more information). You can use this property to customize menu tool properties from a script.

2.3.2.1.3 MnuBands property

Syntax:

*menu.MnuBands(**index**)*

Data type: array of `MnuBand` objects

Description:

This read-only property array returns a `MnuBand` object corresponding to the specified one-based index.

2.3.2.2 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 and popup menus. A `MnuBand` object contains zero or more [MnuTool](#) objects.

MnuBand properties[BandType property](#)[Caption property](#)[Count property](#)[DockingArea property](#)[MnuTools property](#)[Name property](#)[Visible property](#)**2.3.2.2.1 MnuBand properties****2.3.2.2.1.1 BandType property****Syntax:***mnuband*.BandType**Data type:** integer**Description:**

This property is the type of band: 0 = toolbar, 1 = main menu, 2 = popup menu, 3 = status bar, 4 = session bar.

2.3.2.2.1.2 Caption property (MnuBand)**Syntax:***mnuband*.Caption**Data type:** string**Description:**

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

2.3.2.2.1.3 Count property (MnuBand)**Syntax:***mnuband*.Count**Data type:** integer**Description:**

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

2.3.2.2.1.4 DockingArea property**Syntax:***mnuband*.DockingArea**Data type:** integer

Description:

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

2.3.2.2.1.5 MnuTools property**Syntax:**

mnuband.MnuTools(*index*)

Data type: array of MnuTool objects

Description:

This read-only property array returns a MnuTool object corresponding to the specified one-based index.

2.3.2.2.1.6 Name property**Syntax:**

mnuband.Name

Data type: string

Description:

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

2.3.2.2.1.7 Visible property (MnuBand)**Syntax:**

mnuband.Visible

Data type: boolean

Description:

This property is True if the band is visible.

2.3.2.3 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.

MnuTool properties

[Action property](#)

[Caption property](#)

[Checked property](#)

[Enabled property](#)

[ToolID property](#)

[ToolTipText property](#)

[Visible property](#)

2.3.2.3.1 MnuTool properties

2.3.2.3.1.1 Action property

Syntax:

mnutool.Action

Data type: string

Description:

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.

2.3.2.3.1.2 Caption property (MnuTool)

Syntax:

mnutool.Caption

Data type: string

Description:

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

2.3.2.3.1.3 Checked property

Syntax:

mnutool.Checked

Data type: boolean

Description:

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

2.3.2.3.1.4 Enabled property

Syntax:

mnutool.Enabled

Data type: integer

Description:

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.

2.3.2.3.1.5 ToolID property

Syntax:

mnutool.ToolID

Data type: long

Description:

This property is the tool ID.

2.3.2.3.1.6 ToolTipText property

Syntax:

mnutool.ToolTipText

Data type: string

Description:

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

2.3.2.3.1.7 Visible property (MnuTool)

Syntax:

mnutool.Visible

Data type: boolean

Description:

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

2.3.3 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.

See also:

[GetBlock method](#)

[SetBlock method](#)

2.3.4 The 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 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.

Server methods

[AddItem method](#)

[CallSub function](#)
[Connect function](#)
[DeleteItem method](#)
[Disconnect method](#)
[Download method](#)
[Execute function](#)
[Export method](#)
[FileExists function](#)
[IConv function](#)
[Import method](#)
[IsConnected function](#)
[ItemExists function](#)
[OConv function](#)
[ReadItem function](#)
[Readnext function](#)
[UnlockItem method](#)
[Upload method](#)
[WriteItem method](#)

Server properties

[AccountName property](#)
[ErrorMode property](#)
[LastError property](#)
[LastErrorMessage property](#)
[MDName property](#)
[ServerID property](#)
[ServerName property](#)
[UnicodeDelimiters property](#)
[UserName property](#)

2.3.4.1 Server properties

2.3.4.1.1 AccountName property

Syntax:

`Server.AccountName`

Data type: string

Description:

Returns the name of the account the server object is connected to.

2.3.4.1.2 ErrorMode property

Syntax:

`Server.ErrorMode`

Data type: integer

Description:

Set to 0 to ignore errors, 1 to raise an error to the calling application, 2 to display error message dialog box

2.3.4.1.3 LastError property

Syntax:

`Server.LastError`

Data type: integer

Description:

Returns any error number from the previous operation, or zero if no error.

2.3.4.1.4 LastErrorMessage property

Syntax:

`Server.LastErrorMessage`

Data type: string

Description:

Returns any error message from the previous operation, or null if no error.

2.3.4.1.5 MDName property

Syntax:

`Server.MDName`

Data type: string

Description:

Returns the file name of the master dictionary (MD or VOC).

2.3.4.1.6 ServerID property

Syntax:

`Server.ServerID`

Data type: string

Description:

Returns a unique identifier for the server object.

2.3.4.1.7 ServerName property

Syntax:

`Server.ServerName`

Data type: string

Description:

Returns the server name assigned using FTSETUP.

2.3.4.1.8 UnicodeDelimiters property

Syntax:

`Server.UnicodeDelimiters`

Data type: integer

Description:

Set to 1 to convert system delimiter characters to Unicode, or 0 to use traditional character values for system delimiters. When Unicode delimiters are specified, **AM** is U+F8FE, **VM** is U+F8FD and **SVM** is U+F8FC.

2.3.4.1.9 UserName property

Syntax:

`Server.UserName`

Data type: string

Description:

Returns the name of the user logged on to the host.

2.3.4.2 Server methods

2.3.4.2.1 AddItem Method

Syntax:

`Server.AddItem file, ID, data`

Description:

Writes a new item to the Pick database. If the item already exists, error 223 is returned.

2.3.4.2.2 CallSub method

Syntax:

`Result = Server.CallSub (SubName [, function [, data]])`

Data type: string

Description:

Calls a BASIC subroutine (**SubName**) passing **function** and **data** as arguments. The 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.

2.3.4.2.3 Connect method

Syntax:

`Result = Server.Connect([ServerName])`

Data type: boolean

Description:

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.

2.3.4.2.4 DeleteItem method**Syntax:**

```
Server.DeleteItem file, ID
```

Description:

Deletes an item from the Pick database. If the item is locked by another process, error 235 is returned.

2.3.4.2.5 Disconnect method**Syntax:**

```
Server.Disconnect
```

Description:

Disconnects the server object.

2.3.4.2.6 Download method (Server)**Syntax:**

```
Server.Download SourceFile, SourceIDs, TargetFolder [, Protocol [, Binary [, Overwrite ]]]
```

Description:

Uses the AccuTerm FT program to download files from the 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).

2.3.4.2.7 Execute method**Syntax:**

```
Result = Server.Execute(command [, data [, capture ]])
```

Data type: string

Description:

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.

2.3.4.2.8 Export method**Syntax:**

```
Server.Export SourceFile, SourceIDs, TargetFile, TargetFields [, Header [, Explode [, Protocol [, Delimiter [, Overwrite ]]]]]
```

Description:

This method uses the AccuTerm FTD program to download a file from the MultiValue 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).

2.3.4.2.9 FileExists method**Syntax:**

```
Result = Server.FileExists( file )
```

Data type: integer

Description:

Returns 1 if file exists on the host account, or 0 if it does not exist.

2.3.4.2.10 IConv method**Syntax:**

```
Result = Server.IConv( data , code )
```

Data type: string

Description:

Performs an "input conversion" on **data**. Conversion to be performed is **code**.

2.3.4.2.11 Import method**Syntax:**

```
Server.Import SourceFile, TargetFile, Reserved1, Fields [, Header [, Skip [, Autold [, AutoldPrefix [, AutoldStart [, Protocol [, Delimiter [, Overwrite ]]]]]]]]
```

Description:

This method uses the AccuTerm FTD program to upload a file from the PC to the MultiValue 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).

2.3.4.2.12 IsConnected method

Syntax:

```
Result = Server.IsConnected( )
```

Data type: boolean

Description:

Returns TRUE if server object is connected to the host and the host is running the server application; otherwise returns FALSE.

2.3.4.2.13 ItemExists function

Syntax:

```
Result = Server.ItemExists( file, ID )
```

Data type: integer

Description:

Returns 1 if item exists in the specified file, or 0 if it does not exist or the file cannot be opened.

2.3.4.2.14 OConv method

Syntax:

```
Result = Server.OConv( data, code )
```

Data type: string

Description:

Performs an "output conversion" on *data*. Conversion to be performed is *code*.

2.3.4.2.15 ReadItem method

Syntax:

```
Result = Server.ReadItem(file, ID [, attr [, value [, subvalue [, locked ]]]] )
```

Data type: string

Description:

Reads an item from the MultiValue 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 MV/BASIC READU statement). Error 260 is returned if the item was already locked by another process.

2.3.4.2.16 Readnext method

Syntax:

```
Result = Server.Readnext(file, attr [, value [, subvalue ]])
```

Data type: string

Description:

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.

2.3.4.2.17 Upload method (Server)

Syntax:

```
Server.Upload SourceFolder, SourceFiles, TargetFile [, Protocol [, Binary [, Overwrite ]]]
```

Description:

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).

2.3.4.2.18 UnlockItem method

Syntax:

```
Server.UnlockItem [ file, [, ID ]]
```

Description:

Unlocks an item locked by the `ReadItem` function. If **file** and **ID** are null, unlocks all items locked by the process.

2.3.4.2.19 WriteItem method

Syntax:

```
Server.WriteItem file, ID, data [, attr [, value [, subvalue [, KeepLocked ]]]]
```

Description:

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 MV/BASIC WRITEU statement).

2.3.5 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 properties[Ansi8Bit property](#)[AnsiAppCursor property](#)[AnsiAppKeypad property](#)[Answerback property](#)[AsciiDelay property](#)[AsciiEOL property](#)[AutoAnswer property](#)[AutoClose property](#)[BackgroundPictureFile property](#)[BackgroundPictureMode property](#)[BackgroundTransparency property](#)[Baud property](#)[BkspSendsDel](#)[BoldFont property](#)[BreakKeyAction property](#)[BytesIn property](#)[BytesOut property](#)[Caption property](#)[CaptureFileName property](#)[CaptureMode property](#)[CaptureSource property](#)[Changed property](#)[Charset property](#)[ClientCertificateID property](#)[Col property](#)[Color property](#)[Colors\(\) property](#)[Cols property](#)[Connected property](#)[ConnectTimeout property](#)[CopySelMode property](#)[CursorType property](#)[DataBits property](#)[DefaultCaptureDir property](#)[DefaultXferDir property](#)[DefaultXferMode property](#)[DefaultXferOverwrite property](#)[Device property](#)[DeviceLicenseMode property](#)[DialStatus property](#)[DisableAppMode property](#)[Duplex property](#)[ExtCols property](#)[ExtRows property](#)[FileName property](#)[FKeys\(\) property](#)

[FontName property](#)
[FontSize property](#)
[GmodeEnable property](#)
[Handshake property](#)
[Height property](#)
[HistoryRows property](#)
[HostCertificateValidation property](#)
[HostKeyFingerprint property](#)
[HostKeyValidation property](#)
[HostName property](#)
[HostPort property](#)
[HostTermType property](#)
[hWnd property](#)
[Icon property](#)
[InputMode property](#)
[KeepaliveInterval property](#)
[KermitCRC property](#)
[KermitEOL property](#)
[KermitErrRetry property](#)
[KermitInitRetry property](#)
[KermitQuote property](#)
[KermitRept property](#)
[KermitTimeout property](#)
[Left property](#)
[LegibleFont property](#)
[LockBaudRate property](#)
[LockFKeys property](#)
[LockKeyboard property](#)
[MapUpperFKeys property](#)
[Menu property](#)
[MessageRows property](#)
[MouseEnable property](#)
[NoAutoWrap property](#)
[NormCols property](#)
[NormRows property](#)
[OnLine property](#)
[OverrideModemConfig property](#)
[Page property](#)
[Pages property](#)
[Palette\(\) property](#)
[Parity property](#)
[PasteAllowCurlyQuotes property](#)
[PasteEOFChar property](#)
[PasteEOFMode property](#)
[PasteEOLChar property](#)
[PasteEOLMode property](#)
[PasteSuppressFinalEOL property](#)
[Port property](#)
[PrinterColorMode property](#)
[PrinterFontBold property](#)
[PrinterFontItalic property](#)
[PrinterFontName property](#)
[PrinterFontSize property](#)

[PrinterMode property](#)
[PrinterName property](#)
[PrinterOrientation property](#)
[PrinterPaperSize property](#)
[PrinterPaperSource property](#)
[PrinterTimeout property](#)
[PrintJobEject property](#)
[PrintScreenBackground property](#)
[PrintScreenEject property](#)
[ProtectAttr property](#)
[Row property](#)
[Rows property](#)
[ScaleFont property](#)
[ScreenPrintAdapterScript property](#)
[ScreenPrinterName property](#)
[ScreenPrintMode property](#)
[ScreenPrintOrientation property](#)
[ScreenPrintPaperSize property](#)
[ScreenPrintPaperSource property](#)
[ScrMode property](#)
[ScrollMode property](#)
[Selection property](#)
[ShowErrs property](#)
[SlavePrintAdapterScript property](#)
[SlavePrinterName property](#)
[SlavePrintMode property](#)
[Sound property](#)
[SSHAAuth property](#)
[SSHCipherProperty](#)
[SSHKey property](#)
[SSHUseOnlyPreferredCipher property](#)
[SSHVersion property](#)
[StopBits property](#)
[Strip8th property](#)
[Tcpldn property](#)
[TcpNoDelay property](#)
[TelnetBinary property](#)
[TelnetBypass property](#)
[TelnetKeepalivesNOP property](#)
[TermType property](#)
[TextFileEncoding property](#)
[ThemeStyle property](#)
[Top property](#)
[Visible property](#)
[Width property](#)
[WindowState property](#)
[XferBytes property](#)
[XferFiles property](#)
[XferStatus property](#)
[XmodemTimeout property](#)
[XmodemTimeout property](#)
[ZmodemTimeout property](#)

Session methods

[Activate method](#)
[Break method](#)
[Capture method](#)
[CaptureEnd method](#)
[Clear method](#)
[ClearSelection method](#)
[Close method](#)
[Copy method](#)
[CopyHistory method](#)
[Delete method](#)
[Deselect method](#)
[Dial method](#)
[Download method](#)
[Emulate method](#)
[ExtMode method](#)
[GetBlock method](#)
[GetSelection method](#)
[GetText method](#)
[Hangup method](#)
[Hide method](#)
[Input method](#)
[LoadImage method](#)
[Locate method](#)
[MouseTableAdd method](#)
[MouseTableLoad method](#)
[MouseTableReset method](#)
[Move method](#)
[NormMode method](#)
[Output method](#)
[Paste method](#)
[PlayMidi method](#)
[PlayWave method](#)
[PrinterClose method](#)
[PrinterOff method](#)
[PrinterOn method](#)
[PrintScreen method](#)
[ReadText method](#)
[Reset method](#)
[ResetComm method](#)
[ResetTerm method](#)
[Resize method](#)
[Save method](#)
[SaveAs method](#)
[ScrollHistory method](#)
[Select method](#)
[SetBlock method](#)
[SetExtension method](#)
[SetSelection method](#)
[SetText method](#)
[Settings method](#)
[Show method](#)
[Terminate method](#)

[UnloadImage method](#)

[Upload method](#)

[WaitFor method](#)

[WriteText method](#)

Session events

[DataReady event](#)

[Extension event](#)

2.3.5.1 Session properties

2.3.5.1.1 Ansi8Bit property

Syntax:

session.Ansi8Bit

Data type: boolean

Description:

If this setting is non-zero, AccuTerm sends 8 bit control codes. Otherwise the equivalent 7-bit escape sequence is sent. This property is only effective when AccuTerm is emulating one of the VT terminals.

2.3.5.1.2 AnsiAppCursor property

Syntax:

session.AnsiAppCursor

Data type: boolean

Description:

If this setting is non-zero, AccuTerm sends "application codes" instead of "cursor codes" when cursor keys are pressed. This property is only effective when AccuTerm is emulating one of the VT terminals.

2.3.5.1.3 AnsiAppKeypad property

Syntax:

session.AnsiAppKeypad

Data type: boolean

Description:

If this setting is non-zero, AccuTerm sends "application codes" instead of numeric characters when key on the numeric keypad are pressed. This property is only effective when AccuTerm is emulating one of the VT terminals.

2.3.5.1.4 Answerback property

Syntax:

session.Answerback

Data type: string

Description:

This is the string that AccuTerm returns to the host system when the host requests the "answerback message".

2.3.5.1.5 AsciiDelay property**Syntax:**

session.AsciiDelay

Data type: integer

Description:

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

2.3.5.1.6 AsciiEOL property**Syntax:**

session.AsciiEOL

Data type: integer

Description:

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

2.3.5.1.7 AutoAnswer property**Syntax:**

session.AutoAnswer

Data type: boolean

Description:

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

2.3.5.1.8 AutoClose property (Session)**Syntax:**

session.AutoClose

Data type: boolean

Description:

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

2.3.5.1.9 BackgroundPictureFile property**Syntax:**

session.BackgroundPictureFile

Data type: string

Description:

This is the name of an image file that is used for the session background.

See also:

[BackgroundPictureMode property](#)

[BackgroundTransparency property](#)

2.3.5.1.10 BackgroundPictureMode property

Syntax:

session.BackgroundPictureMode

Data type: integer

Description:

This property controls how the background picture is scaled: 0 - picture is stretched to fit screen size. This may alter the picture's aspect ratio; 1 - picture is scaled so that its longest dimension fills the screen. The area not filled by the picture is black; 2 - picture is scaled so that its shortest dimension fills the screen. The longer dimension is cropped.

See also:

[BackgroundPictureFile property](#)

[BackgroundTransparency property](#)

2.3.5.1.11 BackgroundTransparency property

Syntax:

session.BackgroundTransparency

Data type: integer

Description:

The BackgroundTransparency property specifies the transparency of the background color or the opacity of the background picture, as a percentage. 100% means that the picture is used as the background, 0% means that the background color is used. Values in between blend the background picture and background color.

See also:

[BackgroundPictureFile property](#)

[BackgroundPictureMode property](#)

2.3.5.1.12 Baud property

Syntax:

session.Baud

Data type: integer

Description:

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`.

2.3.5.1.13 BkspSendsDel property**Syntax:**

`session.BkspSendsDel`

Data type: boolean

Description:

If this setting is True (non-zero), pressing the Backspace key causes AccuTerm to send the *DEL* control code. Otherwise, AccuTerm sends the *BS* control code.

2.3.5.1.14 BoldFont property**Syntax:**

`session.BoldFont`

Data type: boolean

Description:

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

2.3.5.1.15 BreakKeyAction property**Syntax:**

`session.BreakKeyAction`

Data type: integer

Description:

This property specifies the action to take when the CTRL+BREAK key is pressed. The exact interpretation of this property depends on the connection type.

- 0 - default action - send normal BREAK signal or character
- 1 - no break key
- 2 - send normal break (telnet BREAK command or SSH BREAK command)
- 3 - send alternate break (telnet IP command or SSH SIGINT signal)
- 256+*break_char* - send *break_char* character (ASCII code)

2.3.5.1.16 BytesIn property**Syntax:**

`session.BytesIn`

Data type: long

Description:

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

2.3.5.1.17 BytesOut property**Syntax:**

`session.BytesOut`

Data type: long

Description:

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

2.3.5.1.18 CapsLock property**Syntax:**

`session.CapsLock`

Data type: boolean

Description:

Set to True (non-zero) to set the state of the CapsLock key; set to False (zero) to reset the CapsLock key state.

2.3.5.1.19 Caption property**Syntax:**

`session.Caption`

Data type: string

Description:

Set or returns the session caption (title).

2.3.5.1.20 CaptureFileName property**Syntax:**

`session.CaptureFileName`

Data type: string

Description:

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

2.3.5.1.21 CaptureMode property**Syntax:**

`session.CaptureMode`

Data type: integer

Description:

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). If capture mode has 128 added to it, this indicates that control characters are being filtered from the captured data. This property is read-only.

2.3.5.1.22 CaptureSource property

Syntax:

session.CaptureSource

Data type: integer

Description:

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

2.3.5.1.23 Changed property

Syntax:

session.Changed

Data type: integer

Description:

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.

2.3.5.1.24 Charset property

Syntax:

session.Charset

Data type: string

Description:

The character set used by the host. The only valid values are NULL for the default character set for the selected terminal emulation, or "NATIVE" if the host uses the current 8-bit (non-Unicode) Windows character set.

2.3.5.1.25 ClientCertificateID property

Syntax:

session.ClientCertificateID

Data type: string

Description:

If an SSL/Telnet host requires the client to authenticate using a certificate, this property contains the name and/or hash of the desired client certificate. The client certificate must be installed in the current user's "MY" certificate store.

2.3.5.1.26 Col & Row properties

Syntax:`session.Col``session.Row`**Data type:** integer**Description:**

The current cursor column or row. The leftmost column is zero and the topmost row is zero. Also see the [Locate method](#).

2.3.5.1.27 Color property

Syntax:`session.Color`**Data type:** integer**Description:**

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:

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

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

2.3.5.1.28 Colors() property

Syntax:`session.Colors(index)`

Data type: integer

Description:

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:

Index	Attribute Description	Index	Attribute Description
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

The value of this property may include a border effect style which is associated with the visual attribute. The border effect styles are combined with the foreground and background color as follows:

```

none = fg + bg
inset = fb + bg + 16384
raised = fg + bg - 32768
flat = fg + bg - 16384

```

2.3.5.1.29 Cols & Rows properties

Syntax:

`session.Cols`

`session.Rows`

Data type: integer

Description:

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

2.3.5.1.30 Connected property

Syntax:

`session.Connected`

Data type: boolean

Description:

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.

2.3.5.1.31 ConnectTimeout property

Syntax:

session.ConnectTimeout

Data type: integer

Description:

This property is the number of seconds to wait when attempting to make a connection before a timeout error occurs.

2.3.5.1.32 CopyPasteShortcut property

Syntax:

session.CopyPasteShortcut

Data type: integer

Description:

Set to 0 for no shortcut, 1 to use CTRL+INS / SHIFT+INS, or 2 to use CTRL+C / CTRL+V for Copy / Paste shortcuts.

2.3.5.1.33 CopySelMode property

Syntax:

session.CopySelMode

Data type: integer

Description:

Set to 1 to clear the selection rectangle after copying the screen selection to the clipboard. Set to 0 to retain selection rectangle.

2.3.5.1.34 CursorType property

Syntax:

session.CursorType

Data type: boolean

Description:

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

2.3.5.1.35 CustomMouseTable property (Session)

Syntax:

`session.CustomMouseTable`

Data type: string

Description:

This is the file name of the custom mouse table file for the session. The initial value of this property is the global mouse table file. See [Mouse Pattern Matching](#) for more information on using a custom mouse table.

2.3.5.1.36 DataBits property

Syntax:

`session.DataBits`

Data type: integer

Description:

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 `atDevSERIAL` or `atDevMODEM`. Acceptable values are 7 or 8. After changing this property, use the `Reset` method for the change to take effect.

2.3.5.1.37 DefaultCaptureDir property

Syntax:

`session.DefaultCaptureDir`

Data type: string

Description:

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

2.3.5.1.38 DefaultXferDir property

Syntax:

`session.DefaultXferDir`

Data type: string

Description:

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

2.3.5.1.39 DefaultXferMode property

Syntax:

`session.DefaultXferMode`

Data type: integer

Description:

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

2.3.5.1.40 DefaultXferOverwrite property

Syntax:

session.DefaultXferOverwrite

Data type: integer

Description:

This is default overwrite setting for received files. Set to True (non-zero) to allow overwrites, else set to False (zero). This property only applies to ASCII and Xmodem downloads.

2.3.5.1.41 Device property

Syntax:

session.Device

Data type: integer

Description:

The communications device type attached to the session. The device type may be `atDevNONE` for no device (disconnects session), `atDevSERIAL`, `atDevTELNET`, `atDevSSH` or `atDevMODEM`.

2.3.5.1.42 DeviceLicenseMode property

Syntax:

session.DeviceLicenseMode

Data type: integer

Description:

If this setting is non-zero, "device licensing" negotiation between AccuTerm and the host system is enabled. When device licensing has been successfully negotiated, multiple AccuTerm connections (sessions) to the host system consume only one user license.

To enable device license negotiation, set this value to 1 if the device license response requires a *LF* termination character, or 3 if the response requires no termination character.

Device licensing is supported for UniVerse, UniData and QM host systems. The host system must have device licensing enabled (this is often an extra-charge item).

2.3.5.1.43 DialStatus property

Syntax:

session.DialStatus

Data type: integer

Description:

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

Status	Description
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.

2.3.5.1.44 DisableAppMode property

Syntax:

session.DisableAppMode

Data type: boolean

Description:

Setting this property to True (non-zero) prevents AccuTerm from entering "keypad application mode" or "cursor application mode" when running one of the VT emulations. Set to False to allow application mode.

2.3.5.1.45 Duplex property

Syntax:

session.Duplex

Data type: integer

Description:

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).

2.3.5.1.46 ExtCols & ExtRows properties

Syntax:

session.ExtCols

session.ExtRows

Data type: integer

Description:

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

2.3.5.1.47 ExtMode method

Syntax:

session.ExtMode

Description:

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

2.3.5.1.48 FileName property

Syntax:

session.FileName

Data type: string

Description:

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.

2.3.5.1.49 FKeys() property

Syntax:

session.FKeys(*index*)

Data type: string

Description:

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 special modifier for the function key button bar caption can be combined with one of the standard modifiers to retrieve or store the caption & tooltip text. If both caption and tooltip are used, separate them with a pipe (|) character, caption first (a caption can be used without a tooltip). The string value of the programmed key is binary, and might not contain printable characters.

Modifier	Shift	Ctrl	Alt	Modifier	Shift	Ctrl	Alt
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
10000	retrieve the function key caption and tip string						

Virtual Key	Number	Virtual Key	Number
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

Virtual Key	Number	Virtual Key	Number
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		

See also:

[Settings FKeys\(\) property](#)

2.3.5.1.50 FontName property

Syntax:

session.FontName

Data type: string

Description:

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

2.3.5.1.51 FontSize property

Syntax:

session.FontSize

Data type: single

Description:

This is the size of the terminal font in points.

2.3.5.1.52 GmodeEnable property

Syntax:

session.GmodeEnable

Data type: boolean

Description:

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

2.3.5.1.53 Handshake property

Syntax:

session.Handshake

Data type: integer

Description:

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` or `atDevMODEM`. Acceptable handshake settings are `atHandshakeNONE`, `atHandshakeXON`, `atHandshakeXIO`, `atHandshakeRTS` or `atHandshakeDTR`.

2.3.5.1.54 Height property (Session)

Syntax:

`session.Height`

Data type: long

Description:

This is the session window height in pixels.

2.3.5.1.55 HistoryRows property

Syntax:

`session.HistoryRows`

Data type: integer

Description:

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.

2.3.5.1.56 HostCertificateValidation property

Syntax:

`session.HostCertificateValidation`

Data type: long

Description:

This property determines how the host certificate is validated when connecting using SSL/Telnet. Set this value to 0 if no validation is desired or -1 for strict validation. Otherwise, this is a bit mask with the following values:

- 1 = host certificate must be signed by a trusted certificate authority
- 2 = host certificate name must match host name
- 4 = host certificate must not be expired
- 8 = host certificate must not be revoked
- 16 = certificate must be valid for server authentication

2.3.5.1.57 HostKeyFingerprint property**Syntax:**

session.HostKeyFingerprint

Data type: string

Description:

This property contains the hash (fingerprint) of the SSH host key.

2.3.5.1.58 HostKeyValidation property**Syntax:**

session.HostKeyValidation

Data type: long

Description:

When using an SSH connection, set this property to 0 to ignore the host's public key, 1 if the host key should be trusted if it has not yet been set, 2 to confirm any change in the host key, or 3 if the key must match the key fingerprint specified in the `HostKeyFingerprint` property.

2.3.5.1.59 HostName property**Syntax:**

session.HostName

Data type: string

Description:

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 Telnet or Secure Shell. 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.

2.3.5.1.60 HostPort property**Syntax:**

session.HostPort

Data type: string

Description:

For Telnet connections, this is the host TCP/IP port number assigned for Telnet services (default port is 23). For Secure Shell connections, it is the TCP/IP port assigned for SSH services (default port is 22).

2.3.5.1.61 HostTermType property**Syntax:**

session.HostTermType

Data type: string

Description:

For Telnet and Secure Shell 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.

2.3.5.1.62 hWnd property

Syntax:

session.hWnd

Data type: long

Description:

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.

2.3.5.1.63 Icon property

Syntax:

session.Icon

Data type: string

Description:

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

2.3.5.1.64 ID property

Syntax:

session.ID

Data type: string

Description:

This is the unique session ID.

2.3.5.1.65 InputMode property

Syntax:

session.InputMode

Data type: integer

Description:

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.

2.3.5.1.66 KeepaliveInterval property

Syntax:

session.KeepaliveInterval

Data type: long

Description:

If this setting is non-zero, AccuTerm will send a special "keepalive" message periodically to maintain the connection (some hosts and routers automatically disconnect when they detect an idle connection.)

This setting is in seconds. *Note: enabling keepalives when connecting to older versions of D3/NT may cause the error log on the host to fill up.)*

2.3.5.1.67 KermitCRC property

Syntax:

session.KermitCRC

Data type: integer

Description:

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.

2.3.5.1.68 KermitEOL property

Syntax:

session.KermitEOL

Data type: integer

Description:

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.

2.3.5.1.69 KermitErrRetry property

Syntax:

session.KermitErrRetry

Data type: integer

Description:

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

2.3.5.1.70 KermitInitRetry property

Syntax:

session.KermitInitRetry

Data type: integer

Description:

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

2.3.5.1.71 KermitQuote property**Syntax:**

session.KermitQuote

Data type: integer

Description:

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.

2.3.5.1.72 KermitRept property**Syntax:**

session.KermitRept

Data type: integer

Description:

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.

2.3.5.1.73 KermitTimeout property**Syntax:**

session.KermitTimeout

Data type: integer

Description:

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

2.3.5.1.74 Left property (Session)**Syntax:**

session.Left

Data type: long

Description:

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.

2.3.5.1.75 LegibleFont property**Syntax:**

session.LegibleFont

Data type: boolean

Description:

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

2.3.5.1.76 LockBaudRate property**Syntax:**

session.LockBaudRate

Data type: boolean

Description:

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.

2.3.5.1.77 LockFKeys property**Syntax:**

session.LockFKeys

Data type: integer

Description:

Set to 0 for unlocked function keys (host can reset or reprogram keys), 1 to lock from reset (host can reprogram keys, but not reset them) or 2 to lock from programming (host cannot reset or reprogram function keys).

2.3.5.1.78 LockKeyboard property**Syntax:**

session.LockKeyboard

Data type: boolean

Description:

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

2.3.5.1.79 MapUpperFKeys property**Syntax:**

session.MapUpperFKeys

Data type: boolean

Description:

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.

2.3.5.1.80 Menu property (Session)**Syntax:**

Set *menu_object* = *session*.Menu

Data type: menu object

Description:

This returns a reference to the session Menu object, which may be used to customize AccuTerm's menus and toolbars.

See also:

[Menu Object](#)

2.3.5.1.81 MessageRows property**Syntax:**

session.MessageRows

Data type: integer

Description:

This property sets the number of message rows and status lines for the session. The value must be between 0 and 3. Note that some emulations do not support any status or message rows, and some only support a single status line. Others support up to 3 lines. For more information, see [Status Lines](#).

2.3.5.1.82 MouseEnable property**Syntax:**

session.MouseEnable

Data type: boolean

Description:

Set to True (non-zero) to enable the ESC STX "1" / ESC STX "2" AccuTerm mouse-on command; otherwise set to False (zero).

2.3.5.1.83 NoAutoWrap property**Syntax:**

session.NoAutoWrap

Data type: boolean

Description:

Set to True (non-zero) to disable automatic line wrap at end of line.

2.3.5.1.84 NormCols & NormRows properties

Syntax:`session.NormCols``session.NormRows`**Data type:** integer**Description:**

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

2.3.5.1.85 NormMode method

Syntax:`session.NormMode`**Description:**

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

2.3.5.1.86 OnLine property

Syntax:`session.OnLine`**Data type:** integer**Description:**

Setting this property to True (non-zero) initiates the connection to the host. Setting it to False terminates the connection.

2.3.5.1.87 OverrideModemConfig property

Syntax:`session.OverrideModemConfig`**Data type:** boolean**Description:**

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).

2.3.5.1.88 Page property

Syntax:`session.Page`**Data type:** integer**Description:**

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).

2.3.5.1.89 Pages property

Syntax:

session.Pages

Data type: integer

Description:

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

2.3.5.1.90 Palette() property

Syntax:

session.Palette(*index*)

Data type: OLE_COLOR

Description:

This array contains the color values for the 16 palette entries used to display text on the terminal screen. The [Color](#) property assigns foreground (text) and background colors based on these 16 colors. You can use the RGB() function to produce an OLE_COLOR from separate red, green and blue values. The red, green and blue values must be in the range of 0 to 255.

Palette entry	Color
0	Black
1	Dark Blue
2	Dark Green
3	Turquoise
4	Dark Red
5	Purple
6	Olive
7	Light Grey
8	Dark Grey
9	Blue
10	Green
11	Cyan
12	Red
13	Magenta
14	Yellow
15	White

An additional 6 palette entries exist for drawing borders (see [Visual Styles](#)). These are:

Palette entry	Used for
16	flat outer border
17	flat inner border
18	shadow outer border

Palette entry	Used for
19	shadow inner border
20	highlight outer border
21	highlight inner border

2.3.5.1.91 Parity property

Syntax:

session.Parity

Data type: integer

Description:

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 `atDevSERIAL` or `atDevMODEM`. Acceptable values are `atParityNONE`, `atParityEVEN`, `atParityODD`, `atParityMARK` and `atParitySPACE`. After changing this property, use the `Reset` method for the change to take effect.

2.3.5.1.92 PasteAllowCurlyQuotes property

Syntax:

session.PasteAllowCurlyQuotes

Data type: integer

Description:

Set to 0 (default) if AccuTerm should automatically translate "curly quotes" into "straight quotes" when pasting. Set to 1 to allow "curly quotes" to be pasted unchanged. *Note: "curly quotes" are only supported if your host character set encoding is "Unicode / UTF-8".*

2.3.5.1.93 PasteEOFChar property

Syntax:

session.PasteEOFChar

Data type: integer

Description:

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.

2.3.5.1.94 PasteEOFMode property

Syntax:

session.PasteEOFMode

Data type: integer

Description:

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.

2.3.5.1.95 PasteEOLChar property

Syntax:

session.PasteEOLChar

Data type: integer

Description:

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.

2.3.5.1.96 PasteEOLMode property

Syntax:

session.PasteEOLMode

Data type: integer

Description:

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. If you want to suppress the end-of-line character after the *last* line, add 128 to this value.

2.3.5.1.97 PasteSuppressFinalEOL property

Syntax:

session.PasteSuppressFinalEOL

Data type: integer

Description:

Set to 1 if AccuTerm should omit the EOL character after the last line (or only line) when pasting. Set to 0 (default) to send EOL after each line.

2.3.5.1.98 Port property

Syntax:

session.Port

Data type: integer

Description:

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 127. After changing this property, use the `Reset` method for the change to take effect.

2.3.5.1.99 PrinterColorMode property**Syntax:**

session.PrinterColorMode

Data type: integer

Description:

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).

2.3.5.1.100 PrinterFontBold property**Syntax:**

session.PrinterFontBold

Data type: boolean

Description:

Set this property to True (non-zero) to print using boldface. This property affects slave (or Aux) print jobs when the `SlavePrintMode` property is 1 (Graphics mode). This property has no effect when the `SlavePrintMode` property is zero (Text mode).

2.3.5.1.101 PrinterFontItalic property**Syntax:**

session.PrinterFontItalic

Data type: boolean

Description:

Set this property to True (non-zero) to print using italics. This property affects slave (or Aux) print jobs when the `SlavePrintMode` property is 1 (Graphics mode). This property has no effect when the `SlavePrintMode` property is zero (Text mode).

2.3.5.1.102 PrinterFontName property**Syntax:**

session.PrinterFontName

Data type: string

Description:

Set this property to the name of the font to use for slave (or Aux) print jobs when the `SlavePrintMode` property is 1 (Graphics mode). This property has no effect when the `SlavePrintMode` property is zero (Text mode).

2.3.5.1.103 PrinterFontSize property**Syntax:**

session.PrinterFontSize

Data type: single

Description:

Set this property to the size (in points) of the font to use for slave (or Aux) print jobs when the `SlavePrintMode` property is 1 (Graphics mode). This property has no effect when the `SlavePrintMode` property is zero (Text mode).

2.3.5.1.104 PrinterMode property

Syntax:

`session.PrinterMode`

Data type: integer

Description:

This is the state of the auxiliary printer: 0 if printer is off (`atPrinterOff`), 1 if printer is auto-print mode (`atPrinterOn`), 2 if the printer is in transparent print mode (`atPrinterTransparent`).

2.3.5.1.105 PrinterName property

Syntax:

`session.PrinterName`

`session.SlavePrinterName`

Data type: string

Description:

This is the name of the printer used by the session for slave (or Aux) printer output. It may be set to null ("") to indicate the current Windows default printer, or "no printer" to disable slave printing.

See also:

[ScreenPrinterName property](#)

2.3.5.1.106 PrinterOrientation property

Syntax:

`session.PrinterOrientation`

Data type: integer

Description:

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 1 (Graphics mode). This property has no effect when the `SlavePrintMode` property is zero (Text mode).

2.3.5.1.107 PrinterPaperSize property

Syntax:

`session.PrinterPaperSize`

Data type: integer

Description:

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 3.875 x 7.5 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 1/2 x 13 in	

2.3.5.1.108 PrinterPaperSource property**Syntax:**

session.PrinterPaperSource

Data type: integer

Description:

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 1 (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

2.3.5.1.109 PrinterTimeout property

Syntax:

session.PrinterTimeout

Data type: integer

Description:

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

2.3.5.1.110 PrintJobEject property

Syntax:

session.PrintJobEject

Data type: integer

Description:

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.

2.3.5.1.111 PrintScreenBackground property

Syntax:

session.PrintScreenBackground

Data type: boolean

Description:

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

2.3.5.1.112 PrintScreenEject property

Syntax:

session.PrintScreenEject

Data type: integer

Description:

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.

2.3.5.1.113 ProtectAttr property

Syntax:

session.ProtectAttr

Data type: integer

Description:

Visual attribute number assigned to "protected fields" under Wyse/ADD5 emulation. See [Colors\(\)](#) property for a list of attribute numbers.

2.3.5.1.114 Settings method

Syntax:

```
With session.Settings
    . . . statements
End With
```

Data type: settings object

Description:

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.

Note: when using the Settings object from a .NET language, call the Apply method just before End With due to the way objects are freed in the .NET environment.

2.3.5.1.115 ScaleFont property

Syntax:

session.ScaleFont

Data type: boolean

Description:

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

2.3.5.1.116 ScreenPrintAdapterScript property

Syntax:

session.ScreenPrintAdapterScript

Data type: string

Description:

This is the name of the print adapter script file to use for screen printing. Print adapter scripts are used

to process the print job using an external program, such as Print Wizard. To use the print adapter script, the [ScreenPrintMode](#) property must be set to 2. See [Screen Printer Settings](#) for more information on print adapters.

2.3.5.1.117 ScreenPrinterName property

Syntax:

session.ScreenPrinterName

Data type: string

Description:

This is the name of the printer used by the session for screen printing. It may be set to null ("") to indicate the current Windows default printer, or "no printer" to disable screen printing.

See also:

[SlavePrinterName property](#)

2.3.5.1.118 ScreenPrintMode property

Syntax:

session.ScreenPrintMode

Data type: integer

Description:

Set to 0 for text mode, 1 for graphics mode or 2 to use the print adapter script specified by the [ScreenPrintAdapterScript](#) property. When this property is 1, the [PrinterColorMode](#) and [PrintScreenBackground](#) properties affect how the screen is printed.

2.3.5.1.119 ScreenPrintOrientation property

Syntax:

session.ScreenPrintOrientation

Data type: integer

Description:

Set this property to 0 use the default page orientation, 1 for portrait or 2 for landscape. This property applies only to screen printing and only when the [ScreenPrintMode](#) property is 1 (Graphics mode). This property has no effect when the [ScreenPrintMode](#) property is zero (Text mode).

2.3.5.1.120 ScreenPrintPaperSize property

Syntax:

session.ScreenPrintPaperSize

Data type: integer

Description:

Set this property to 0 use the default paper size for the printer. Otherwise use a paper size from the table shown in the [PrinterPaperSize](#) property. This property applies only when the [ScreenPrintMode](#)

property is non-zero (Graphics mode). This property has no effect when the `ScreenPrintMode` property is zero (Text mode).

2.3.5.1.121 ScreenPrintPaperSource property

Syntax:

session.`ScreenPrintPaperSource`

Data type: integer

Description:

Set this property to 0 use the default paper source for the printer. Otherwise use a paper source from the table shown in the [PrinterPaperSource](#) property. This property applies only to screen printing and only when the [ScreenPrintMode](#) property is 1 (Graphics mode). This property has no effect when the `ScreenPrintMode` property is zero (Text mode).

2.3.5.1.122 ScrMode property

Syntax:

session.`ScrMode`

Data type: integer

Description:

Set to zero for normal (80 column) mode, 1 for extended (132 column) mode, or 2 for variable (programmers mode). *Note: when using variable mode, you must use a fixed size font.*

2.3.5.1.123 ScrollMode property

Syntax:

session.`ScrollMode`

Data type: integer

Description:

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.

2.3.5.1.124 Selection property

Syntax:

session.`Selection`

Data type: string

Description:

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

See also:

[Select method](#)

[GetSelection method](#)
[SetSelection method](#)

2.3.5.1.125 ShowErrs property

Syntax:

`session.ShowErrs`

Data type: boolean

Description:

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

2.3.5.1.126 SlavePrintAdapterScript property

Syntax:

`session.SlavePrintAdapterScript`

Data type: string

Description:

This is the name of the print adapter script file to use for slave (Aux) print jobs. Print adapter scripts are used to process the print job using an external program, such as Print Wizard. To use the print adapter script, the [SlavePrintMode](#) property must be set to 2. See [Slave Printer Settings](#) for more information on print adapters.

2.3.5.1.127 SlavePrinterName property

Syntax:

`session.PrinterName`

`session.SlavePrinterName`

Data type: string

Description:

This is the name of the printer used by the session for slave (or Aux) printer output. It may be set to null ("") to indicate the current Windows default printer, or "no printer" to disable slave printing.

See also:

[ScreenPrinterName property](#)

2.3.5.1.128 SlavePrintMode property

Syntax:

`session.SlavePrintMode`

Data type: integer

Description:

Set to 0 for text mode, 1 for graphics mode or 2 to use the print adapter script specified in the [SlavePrintAdapterScript](#) property. When this property is 1, the `PrinterFont...`, `PrinterOrientation`, `PrinterPaperSize` and `PrinterPaperSource` properties affect how the slave (Aux) print job is printed.

Print adapter scripts are used to process the print job using an external program, such as PrintWizard. See [Slave Printer Settings](#) for more information on print adapters.

2.3.5.1.129 Sound property

Syntax:

session.Sound

Data type: string

Description:

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 WAVE, MP3 or MIDI 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).

2.3.5.1.130 SSHAuth property

Syntax:

session.SSHAuth

Data type: integer

Description:

This property may be used to select the authentication method which will be used by a Secure Shell (ssh) session to authenticate a user to the host. Available authentication methods are: public key (2), password (3) and interactive (7). Use zero (0) for the default authentication method (password).

2.3.5.1.131 SSHCipher property

Syntax:

session.SSHCipher

Data type: integer

Description:

This property may be used to select a cipher to be used to encrypt a Secure Shell (ssh) session. Available ciphers are: Triple DES (3), Blowfish (6), 128 bit AES (7) and 256 bit AES (8). Use zero (0) for the default cipher (Triple DES for SSH1 or 128 bit AES for SSH2).

2.3.5.1.132 SSHKey property

Syntax:

session.SSHKey

Data type: string

Description:

This property contains the name of the SSH private key file. If no path is included, AccuTerm's default key folder will be used to read the key file.

2.3.5.1.133 SSHUseOnlyPreferredCipher property

Syntax:

session.SSHUseOnlyPreferredCipher

Data type: boolean

Description:

Set to True (non-zero) to require that the cipher used for an SSH connection is the cipher specified by the [SSHCipher](#) property. If the host system does not accept the preferred cipher, the connection is aborted.

2.3.5.1.134 SSHVersion property

Syntax:

session.SSHVersion

Data type: integer

Description:

This property may be used to select the desired SSH protocol version. Set this property to zero (0) for automatic protocol selection. Set to 1 for SSH version 1 or set to 2 for SSH version 2. When automatic protocol selection is used, AccuTerm selects the highest protocol version supported by the host.

2.3.5.1.135 StopBits property

Syntax:

session.StopBits

Data type: integer

Description:

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 `atDevSERIAL` or `atDevMODEM`. Acceptable values are 1 or 2. After changing this property, use the `Reset` method for the change to take effect.

2.3.5.1.136 Strip8th property

Syntax:

session.Strip8th

Data type: boolean

Description:

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

2.3.5.1.137 TcpIdn property

Syntax:

session.TcpIdn

Data type: boolean

Description:

Set this property to True (non-zero) to enable Internationalized Domain Names (IDN) for the host name. IDN relies upon a standardized mechanism known as "Punycode" for encoding Unicode domain names using only the ASCII characters that are permitted by the global DNS system.

2.3.5.1.138 TcpNoDelay property

Syntax:

session.TcpNoDelay

Data type: boolean

Description:

Normally, the TCP network stack delays outbound network messages for a short time in order to "coalesce" multiple small messages into a single packet. When this setting is True (non-zero), AccuTerm will send each message as soon as possible. Setting this option may improve throughput, especially when running in AccuTerm's GUI environment and during file transfers, at the expense of increased network traffic.

2.3.5.1.139 TelnetBinary property

Syntax:

session.TelnetBinary

Data type: boolean

Description:

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

2.3.5.1.140 TelnetBypass property

Syntax:

session.TelnetBypass

Data type: boolean

Description:

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 D3 which do not implement the complete Telnet protocol.

2.3.5.1.141 TelnetKeepaliveIsNOP property

Syntax:

session.TelnetKeepaliveIsNOP

Data type: boolean

Description:

Set to True (non-zero) to send Telnet NOP command for keepalives; otherwise, sends Telnet DMARK command for keepalives.

2.3.5.1.142 TermType property

Syntax:

session.TermType

Data type: integer

Description:

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
atTermVT100	DEC VT-100
atTermVT220	DEC VT-220
atTermVT320	DEC VT-320
atTermVT420	DEC VT-420
atTermLinux	Linux console
atTermSCO	SCO console
atTermANSI	ANSI BBS
atTermPICKMON	Pick PC Monitor
atTermP30	MDIS P30
atTermTEK	Tektronix 4014
atTermTEK4100	Tektronix 4100

2.3.5.1.143 TextFileEncoding property

TextFileEncoding property

Syntax:

session.TextFileEncoding

Data type: integer

Description:

Sets the encoding to use for downloaded or captured text files. Set to 1 for ANSI, 2 for Unicode, 3 for UTF-8 or 0 for raw (no encoding).

2.3.5.1.144 ThemeStyle property**Syntax:**

`session.ThemeStyle`

Data type: string

Description:

This property contains the name of the current color theme, or "custom" if using customized colors.

2.3.5.1.145 Top property (Session)**Syntax:**

`session.Top`

Data type: long

Description:

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

2.3.5.1.146 Visible property (Session)**Syntax:**

`session.Visible`

Data type: boolean

Description:

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

2.3.5.1.147 Width property (Session)**Syntax:**

`session.Width`

Data type: long

Description:

This is the session window width in pixels.

2.3.5.1.148 WindowState property (Session)**Syntax:**

`session.WindowState`

Data type: integer

Description:

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

2.3.5.1.149 XferBytes property**Syntax:****`session.XferBytes`****Data type:** long**Description:**

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

2.3.5.1.150 XferFiles property**Syntax:****`session.XferFiles`****Data type:** long**Description:**

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

2.3.5.1.151 XferStatus property**Syntax:****`session.XferStatus`****Data type:** integer**Description:**

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

Status	Description
-1	File transfer in progress.
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.
10	Destination file is write protected.

2.3.5.1.152 XmodemTimeout property**Syntax:****`session.XmodemTimeout`****Data type:** integer

Description:

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

2.3.5.1.153 YmodemTimeout property**Syntax:**

session.YmodemTimeout

Data type: integer

Description:

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

2.3.5.1.154 ZmodemTimeout property**Syntax:**

session.ZmodemTimeout

Data type: integer

Description:

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

2.3.5.2 Session methods**2.3.5.2.1 Activate method (Session)****Syntax:**

session.Activate

Description:

Makes the session the "active" session. There is no return value.

2.3.5.2.2 Break method**Syntax:**

session.Break

Description:

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

2.3.5.2.3 Capture method**Syntax:**

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

Description:

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. Add 128 to **mode** to capture text only (filters out control characters). If **source** is zero or **mode** is 3 or 131, then **filename** is ignored, otherwise it is the name of the file where captured

data is saved.

2.3.5.2.4 CaptureEnd method

Syntax:

session.CaptureEnd

Description:

Terminates any capture operation and closes the capture file.

2.3.5.2.5 Clear method

Syntax:

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

Description:

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](#).

2.3.5.2.6 ClearSelection method

Syntax:

session.ClearSelection

Description:

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

This method is the same as the **Deselect** method.

2.3.5.2.7 Close method (Session)

Syntax:

session.Close [*prompt*]

Description:

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. If **prompt** is zero, the session is closed without prompting.

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

2.3.5.2.8 Copy method

Syntax:

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

Description:

Copies a block of text from session screen to the clipboard. Default coordinates are the upper left and lower right corners of the screen or the current selection if one is present. Negative row numbers will

copy from the history buffer; row -1 is the last history row, etc.

2.3.5.2.9 CopyHistory method

Syntax:

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

Description:

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

2.3.5.2.10 Delete method

Syntax:

session.Delete

Description:

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

2.3.5.2.11 Deselect method

Syntax:

session.Deselect

Description:

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

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

2.3.5.2.12 Dial method

Syntax:

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

Description:

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. This method only works if the **Device** property is **atDevModem**.

2.3.5.2.13 Download method

Syntax:

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

Description:

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.

2.3.5.2.14 Emulate method

Syntax:

session.Emulate *text*

Description:

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

2.3.5.2.15 GetBlock method

Syntax:

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

Description:

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`.

See also:

[ScreenBlock object](#)

2.3.5.2.16 GetSelection method

Syntax:

sel = **session**.GetSelection (*left* , *top* , *right* , *bottom*)

Description:

Gets the coordinates of the selection rectangle on the session screen. The return value *sel* is True (non-zero) if a selection is present; otherwise it is False (zero).

See also:

[SetSelection method](#)

[Selection property](#)

2.3.5.2.17 GetText method

Syntax:

text = **session**.GetText ([*col* [, *row* [, *cols* [, *mode*]]])

Description:

Returns the text contents of a row on the terminal screen. Default *col* and *row* is the current cursor location. Default number of columns is to the end of the row. Use zero for *mode* to return all normal characters on the row, including protected characters. Use 1 to return non-protected characters only. Add 2 to include line-drawing character and special symbols (Unicode). Non-negative row numbers refer to the active (non-history) area of the terminal screen (zero is the first data row of the active screen).

Negative row numbers refer to history rows (row -1 is the last history row).

2.3.5.2.18 Hangup method

Syntax:

result = **session**.Hangup()

Description:

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.

2.3.5.2.19 Hide method (Session)

Syntax:

session.Hide

Description:

This method makes the session window invisible.

2.3.5.2.20 Input method

Syntax:

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

Description:

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.

When using this method, you should set the [InputMode](#) property to 1 or 2 to ensure all received data is returned by this function.

If timeout is negative, the function returns any data that has been received from the host but does not check for more incoming data. This improves performance when used in conjunction with the [DataReady](#) event.

This method is not available when using late binding; use the [ReadText](#) method instead.

2.3.5.2.21 LoadImage method

Syntax:

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

Description:

Creates an image on the terminal screen at the location specified by **col** and **row**. **Filename** specifies the name or URL of the image file. Image file formats supported are BMP, JPEG, GIF, PCD, PCX, PICT, PNG, PSD, TARGA, TIFF, WBMP, XBM, XPM and Windows Metafile. 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.

2.3.5.2.22 Locate method

Syntax:

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

Description:

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

2.3.5.2.23 MouseTableAdd method

Syntax:

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

Description:

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. **Dblclick** is the response if the button is double-clicked over the pattern. See [Mouse Pattern Matching](#) for more information on using a custom mouse table.

2.3.5.2.24 MouseTableLoad method

Syntax:

session.MouseTableLoad *filename*

Description:

Loads a mouse pattern table from a file. See [Mouse Pattern Matching](#) for more information on using a custom mouse table.

2.3.5.2.25 MouseTableReset method

Syntax:

session.MouseTableReset

Description:

Clears the mouse pattern table.

2.3.5.2.26 Move method (Session)

Syntax:

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

Description:

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

2.3.5.2.27 Output method**Syntax:**

session.Output *expr*

Description:

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.

2.3.5.2.28 Paste method**Syntax:**

session.Paste [*filename*]

Description:

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

2.3.5.2.29 PlayMidi method**Syntax:**

session.PlayMidi *filename*

Description:

Plays the media file specified by ***filename***. ***Filename*** may specify a local file or a URL. WAVE, MP3 and MIDI media files are supported.

2.3.5.2.30 PlayWave method**Syntax:**

session.PlayWave *filename*

Description:

Plays the media file specified by ***filename***. ***Filename*** may specify a local file or a URL. WAVE, MP3 and MIDI media files are supported.

2.3.5.2.31 PrinterClose method**Syntax:**

session.PrinterClose

Description:

Closes the current print job.

2.3.5.2.32 PrinterOff method

Syntax:

`session.PrinterOff`

Description:

Turns off the slave (or Aux) printer.

2.3.5.2.33 PrinterOn method

Syntax:

`session.PrinterOn [mode]`

Description:

Turns on the slave (or Aux) 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).

2.3.5.2.34 PrintScreen method

Syntax:

`session.PrintScreen`

Description:

Prints the text screen to the printer.

2.3.5.2.35 ReadText method

Syntax:

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

Description:

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.

When using this method, you should set the [InputMode](#) property to 1 or 2 to ensure all received data is returned by this function.

If timeout is negative, the function returns any data that has been received from the host but does not check for more incoming data. This improves performance when used in conjunction with the [DataReady](#) event.

This method is the same as the [Input](#) method, and is available when using late binding.

2.3.5.2.36 Reset method

Syntax:

`session.Reset [mode]`

Description:

Resets the terminal (*mode* = 1 or `atResetTerminal`), the communication device (*mode* = 2 or `atResetComm`), both (default, *mode* = 0 or `atResetAll`) or closes the connection (*mode* = 3 or `atCloseComm`). Note: you may need to reset the communication device after changing certain device related properties in order for the changes to take effect.

2.3.5.2.37 ResetComm method**Syntax:**

session.ResetComm

Description:

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.*

2.3.5.2.38 ResetTerm method**Syntax:**

session.ResetTerm

Description:

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

2.3.5.2.39 Resize method (Session)**Syntax:**

session.Resize *width, height*

Description:

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

2.3.5.2.40 Save method**Syntax:**

session.Save [*filename*]

Description:

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.

2.3.5.2.41 SaveAs method**Syntax:**

session.SaveAs

Description:

Saves the session configuration prompting for file name.

2.3.5.2.42 ScrollHistory method

Syntax:

session.ScrollHistory *cmd*

Description:

This method manipulates the history scroll position. **Cmd** = atScrlHistLineUp (0) to scroll the history up one line; atScrlHistLineDown (1) to scroll the history down one line, atScrlHistPageUp (2) to scroll up one page, atScrlHistPageDown (3) to scroll down one page, atScrlHistTop (6) to scroll to the top (first line) of the history, atScrlHistBottom (7) to scroll to the bottom (live screen).

2.3.5.2.43 Select method

Syntax:

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

Description:

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

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

See also:

[Selection property](#)

2.3.5.2.44 SetBlock method

Syntax:

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

Description:

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".

See also:

[ScreenBlock object](#)

2.3.5.2.45 SetExtension method

Syntax:

session.SetExtension *leadin* , *terminator*

Description:

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.*

2.3.5.2.46 SetSelection method

Syntax:

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

Description:

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

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

See also:

[GetSelection method](#)

[Selection property](#)

2.3.5.2.47 SetText method

Syntax:

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

Description:

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.

2.3.5.2.48 Show method (Session)

Syntax:

session.Show

Description:

Makes the session window visible.

2.3.5.2.49 Terminate method (Session)

Syntax:

session.Terminate [*prompt*]

Description:

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.

2.3.5.2.50 UnloadImage method

Syntax:

session.UnloadImage [*filename*]

Description:

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.

2.3.5.2.51 Upload method

Syntax:

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

Description:

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.

2.3.5.2.52 WaitFor method

Syntax:

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

Description:

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.

For the WaitFor method to work properly, the session InputMode property should be set to 1 or 2. Otherwise, the terminal emulation process may process characters received from the host before the WaitFor function has a chance to examine them.

2.3.5.2.53 WriteText method

Syntax:

session.WriteText *expr*

Description:

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.

2.3.5.3 Session events

2.3.5.3.1 DataReady event

Syntax:

object.DataReady()

Description:

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.

2.3.5.3.2 Extension event

Syntax:

object.Extension(*text* as String)

Description:

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.

2.3.6 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:

```
Set obj = 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 methods

[Sessions.Add method](#)

[Sessions.Count method](#)

2.3.6.1 Sessions methods

2.3.6.1.1 Add method

Syntax:

Set **object** = Sessions.Add ([*filename* [, *state* [, *mode* [, *hidden*]]]])

Description:

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

2.3.6.1.2 AddClone method

Syntax:

```
Set object = Sessions.AddClone (session [, state [,  
    mode [, hidden]]])
```

Description:

The `Sessions.AddClone` method creates and initializes a new session from an existing session. The return value is a `Session` object. Except for the **session** argument, arguments are optional:

session	source session from which the clone is made.
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

2.3.6.1.3 Count method

Syntax:

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

Description:

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

2.3.7 The Settings Object

The `Settings` method 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. Certain properties of the `Settings` object (and all methods of this object) should *not* be used in scripts or in automation. These properties and methods are noted, and they should only be used when using the AccuTerm Emulator Control in another (custom) application.

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

Note: when using the `Settings` object from a .NET language, call the `Apply` method just before `End With` due to the way objects are freed in the .NET environment.

Settings properties

[Ansi8Bit property](#)

[AnsiAppCursor property](#)

[AnsiAppKeypad property](#)

[Answerback property](#)

[AsciiDelay property](#)

[AsciiEOL property](#)

[AttributeMask property](#)

[AutoAnswer property](#)

[BackgroundPictureFile property](#)

[BackgroundPictureMode property](#)
[BackgroundTransparency property](#)
[Baud property](#)
[BkspSendsDel property](#)
[BoldFont property](#)
[BreakKeyAction property](#)
[Changed property](#) *(do not use in scripts)*
[Charset property](#)
[ClientCertificateID property](#)
[Colors\(\) property](#)
[ConnectTimeout property](#)
[CopySelMode property](#)
[CursorType property](#)
[DataBits property](#)
[DefaultCaptureDir property](#)
[DefaultTextEncoding property](#)
[DefaultXferDir property](#)
[DefaultXferMode property](#)
[DefaultXferOverwrite property](#)
[Device property](#)
[DeviceLicenseMode property](#)
[DisableAppMode property](#)
[Duplex property](#)
[ExtCols property](#)
[ExtRows property](#)
[FKeys\(\) property](#)
[FontName property](#)
[FontSize property](#)
[GmodeEnable property](#)
[Handshake property](#)
[HistoryRows property](#)
[HostCertificateValidation property](#)
[HostKeyFingerprint property](#)
[HostKeyValidation property](#)
[HostName property](#)
[HostPort property](#)
[HostTermType property](#)
[KeepaliveInterval property](#)
[KermitCRC property](#)
[KermitEOL property](#)
[KermitErrRetry property](#)
[KermitInitRetry property](#)
[KermitQuote property](#)
[KermitRept property](#)
[KermitTimeout property](#)
[LegibleFont property](#)
[LockBaudRate property](#)
[LockFKeys property](#)
[MapUpperFKeys property](#)
[MessageRows property](#)
[MouseEnable property](#)
[NoAutoWrap property](#)
[NormCols property](#)

[NormRows property](#)
[OverrideModemConfig property](#)
[Pages property](#)
[Palette\(\) property](#)
[Parity property](#)
[PasteEOFChar property](#)
[PasteEOFMode property](#)
[PasteEOLChar property](#)
[PasteEOLMode property](#)
[PasteAllowCurlyQuotes property](#)
[Port property](#)
[PrinterColorMode property](#)
[PrinterFontBold property](#)
[PrinterFontItalic property](#)
[PrinterFontName property](#)
[PrinterFontSize property](#)
[PrinterName property](#)
[PrinterOrientation property](#)
[PrinterPaperSize property](#)
[PrinterPaperSource property](#)
[PrinterTimeout property](#)
[PrintJobEject property](#)
[PrintScreenBackground property](#)
[PrintScreenEject property](#)
[ProtectAttr property](#)
[ScaleFont property](#)
[ScreenPrintAdapterScript property](#)
[ScreenPrinterName property](#)
[ScreenPrintMode property](#)
[ScreenPrintOrientation property](#)
[ScreenPrintPaperSize property](#)
[ScreenPrintPaperSource property](#)
[ScrMode property](#)
[ScrollMode property](#)
[SlavePrintAdapterScript property](#)
[SlavePrinterName property](#)
[SlavePrintMode property](#)
[Sound property](#)
[SSHAAuth property](#)
[SSHCipher property](#)
[SSHUseOnlyPreferredCipher property](#)
[SSHVersion property](#)
[StopBits property](#)
[Strip8th property](#)
[TcpIcn property](#)
[TcpNoDelay property](#)
[TelnetBinary property](#)
[TelnetBypass property](#)
[TelnetKeepalivesNOP property](#)
[TermType property](#)
[ThemeStyle property](#)
[XmodemTimeout property](#)
[YmodemTimeout property](#)

[ZmodemTimeout property](#)

Settings methods *(do not use in scripts)*

[Apply method](#)

[Initialize method](#)

[Dialog method](#)

[toJSON method](#)

[fromJSON method](#)

2.3.7.1 Settings methods

2.3.7.1.1 Apply method

do not use in scripts

Syntax:

settings.Apply

Description:

The Apply method applies any changed properties to the emulator.

2.3.7.1.2 Dialog method

do not use in scripts

Syntax:

settings.Dialog [**Title**, [**InitSel**, [**Categories**]]]

Description:

This method displays the emulator 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 item **InitSel**, and a list of **Categories** which are displayed. The **Categories** list is a string with a single letter for each category to show in the dialog. If the **categories** argument is null, all categories are shown. Sub-categories for Connection (D1...D7), Colors (C0...C2) and Printer (P1...P2) are also possible, if only a specific settings panel is desired.

Category code	Initial selection	Category
O		General Options
A		Files & Folders
D	3	Connection
T	4	Term Type
S	5	Screen
C	7	Colors & Palette
B		Background
K	8	Keyboard
F	6	Font & Character Set
P	1	Printer
Q		Clipboard
H		Mouse

Category code	Initial selection	Category
E		Sound
X	2	File Transfer
W		Windows
R		TCL Rules

See also:

Using the Settings dialog

2.3.7.1.3 Initialize method

do not use in scripts

Syntax:

settings.Initialize

Description:

This method returns all emulator settings to their default values.

2.3.7.1.4 Settings_toJSON**Syntax:**

settings.toJSON(*formatted*)

Description:

This function returns the emulator settings as a JSON string. Specify *formatted* as 0 to return an unformatted string, or 1 to return a formatted string.

2.3.7.1.5 Settings_fromJSON

Enter topic text here.

2.3.7.2 Settings properties**2.3.7.2.1 Ansi8Bit property****Syntax:**

session.Ansi8Bit

Data type: boolean

Description:

If this setting is non-zero, AccuTerm sends 8 bit control codes. Otherwise the equivalent 7-bit escape sequence is sent. This property is only effective when AccuTerm is emulating one of the VT terminals.

2.3.7.2.2 AnsiAppCursor property**Syntax:**

session.AnsiAppCursor

Data type: boolean

Description:

If this setting is non-zero, AccuTerm sends "application codes" instead of "cursor codes" when cursor keys are pressed. This property is only effective when AccuTerm is emulating one of the VT terminals.

2.3.7.2.3 AnsiAppKeypad property

Syntax:

session.AnsiAppKeypad

Data type: boolean

Description:

If this setting is non-zero, AccuTerm sends "application codes" instead of numeric characters when key on the numeric keypad are pressed. This property is only effective when AccuTerm is emulating one of the VT terminals.

2.3.7.2.4 AnsiAutoprint property

Syntax:

session.AnsiAutoprint

Data type: boolean

Description:

When this setting is True (non-zero), the "auto print" slave printer function works the same as a real VT terminal. That is, when AccuTerm is in "auto print" mode, the entire screen line that the cursor is on is printed when AccuTerm receives a CR, LF or FF control code. If this option is False (zero), then text is printed as it is received from the host (similar to Wyse auto-print mode).

2.3.7.2.5 Answerback property

Syntax:

session.Answerback

Data type: string

Description:

This is the string that AccuTerm returns to the host system when the host requests the "answerback message".

2.3.7.2.6 AsciiDelay property

Syntax:

session.AsciiDelay

Data type: integer

Description:

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

2.3.7.2.7 AsciiEOL property

Syntax:

`session.AsciiEOL`

Data type: integer

Description:

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

2.3.7.2.8 AttributeMask property

Syntax:

`settings.AttributeMask`

Data type: integer

Description:

The AttributeMask property indicates which visual effects are to be displayed. Use 2 for blink, 8 for underline, 10 for both blink and underline or zero to disable both blinking and underline. The AttributeMask property also controls whether border effects are available, as well as the border size. To enable border effects, add 64 for thin internal borders, 128 for thick internal borders, 576 for thin external borders and 640 for thick external borders. When internal borders are selected, the vertical border lines are drawn inside the character cell; when external borders are selected, the vertical border lines are drawn in adjacent cells.

2.3.7.2.9 AutoAnswer property

Syntax:

`session.AutoAnswer`

Data type: boolean

Description:

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

2.3.7.2.10 BackgroundPictureFile property

Syntax:

`session.BackgroundPictureFile`

Data type: string

Description:

This is the name of an image file that is used for the session background.

See also:

[BackgroundPictureMode property](#)

[BackgroundTransparency property](#)

2.3.7.2.11 BackgroundPictureMode property

Syntax:

session.BackgroundPictureMode

Data type: integer

Description:

This property controls how the background picture is scaled: 0 - picture is stretched to fit screen size. This may alter the picture's aspect ratio; 1 - picture is scaled so that its longest dimension fills the screen. The area not filled by the picture is black; 2 - picture is scaled so that its shortest dimension fills the screen. The longer dimension is cropped.

See also:

[BackgroundPictureFile property](#)

[BackgroundTransparency property](#)

2.3.7.2.12 BackgroundTransparency property

Syntax:

session.BackgroundTransparency

Data type: integer

Description:

The BackgroundTransparency property specifies the transparency of the background color or the opacity of the background picture, as a percentage. 100% means that the picture is used as the background, 0% means that the background color is used. Values in between blend the background picture and background color.

See also:

[BackgroundPictureFile property](#)

[BackgroundPictureMode property](#)

2.3.7.2.13 Baud property

Syntax:

session.Baud

Data type: integer

Description:

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.

2.3.7.2.14 BkspSendsDel property

Syntax:

session.BkspSendsDel

Data type: boolean

Description:

If this setting is True (non-zero), pressing the Backspace key causes AccuTerm to send the *DEL* control code. Otherwise, AccuTerm sends the *BS* control code.

2.3.7.2.15 BoldFont property

Syntax:

session.BoldFont

Data type: boolean

Description:

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

2.3.7.2.16 BreakKeyAction property

Syntax:

session.BreakKeyAction

Data type: integer

Description:

This property specifies the action to take when the **CTRL+BREAK** key is pressed. The exact interpretation of this property depends on the connection type.

- 0 - default action - send normal BREAK signal or character
- 1 - no break key
- 2 - send normal break (telnet BREAK command or SSH BREAK command)
- 3 - send alternate break (telnet IP command or SSH SIGINT signal)
- 256+**break_char** - send **break_char** character (ASCII code)

2.3.7.2.17 Changed property (Settings)

do not use in scripts

Syntax:

settings.Changed

Data type: integer

Description:

Non-zero if the settings have been modified.

2.3.7.2.18 CharacterMapping property

Syntax:

session.CharacterMapping

Data type: string

Description:

This property may be used to specify custom character mappings. Use this property to specify the ASCII character value of the Euro symbol as used by your host computer. For example:

```
session.CharacterMapping = "EURO SIGN=128"
```

You can assign any ASCII value for the Euro symbol using this property.

This property can also be used to specify the extended character set mapping for the Prism P30 emulation. To specify the Multinational character set, use:

```
session.CharacterMapping = "MDIS_P30=MULTINATIONAL"
```

To specify the Standard National Extension character set, use:

```
session.CharacterMapping = "MDIS_P30=STANDARD"
```

If you need to specify multiple character mappings, separate each with *CR LF*.

2.3.7.2.19 Charset property

Syntax:

```
session.Charset
```

Data type: string

Description:

The character set used by the host. The only valid values are NULL for the default character set for the selected terminal emulation, or "NATIVE" if the host uses the current 8-bit (non-Unicode) Windows character set.

2.3.7.2.20 ClientCertificateID property

Syntax:

```
session.ClientCertificateID
```

Data type: string

Description:

If an SSL/Telnet host requires the client to authenticate using a certificate, this property contains the name and/or hash of the desired client certificate. The client certificate must be installed in the current user's "MY" certificate store.

2.3.7.2.21 Colors() property

Syntax:

```
session.Colors( index )
```

Data type: integer

Description:

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:

Index	Attribute Description	Index	Attribute Description
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

The value of this property may include a border effect style which is associated with the visual attribute. The border effect styles are combined with the foreground and background color as follows:

```

none = fg + bg
inset = fb + bg + 16384
raised = fg + bg - 32768
flat = fg + bg - 16384

```

2.3.7.2.22 ConnectTimeout property

Syntax:

`session.ConnectTimeout`

Data type: integer

Description:

This property is the number of seconds to wait when attempting to make a connection before a timeout error occurs.

2.3.7.2.23 CopySelMode property

Syntax:

`session.CopySelMode`

Data type: integer

Description:

Set to 1 to clear the selection rectangle after copying the screen selection to the clipboard. Set to 0 to retain selection rectangle.

2.3.7.2.24 CursorType property**Syntax:**

session.CursorType

Data type: boolean

Description:

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

2.3.7.2.25 DataBits property**Syntax:**

session.DataBits

Data type: integer

Description:

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 `atDevSERIAL` or `atDevMODEM`. Acceptable values are 7 or 8. After changing this property, use the `Reset` method for the change to take effect.

2.3.7.2.26 DefaultCaptureDir property**Syntax:**

session.DefaultCaptureDir

Data type: string

Description:

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

2.3.7.2.27 DefaultTextEncoding property**DefaultTextEncoding property**

Syntax:

settings.DefaultTextEncoding

Data type: integer

Description:

Sets the default encoding to use for downloaded or captured text files. Set to 1 for ANSI, 2 for Unicode, 3 for UTF-8 or 0 for raw (no encoding).

2.3.7.2.28 DefaultXferDir property**Syntax:**

session.DefaultXferDir

Data type: string

Description:

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

2.3.7.2.29 DefaultXferMode property

Syntax:

`session.DefaultXferMode`

Data type: integer

Description:

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

2.3.7.2.30 DefaultXferOverwrite property

Syntax:

`session.DefaultXferOverwrite`

Data type: integer

Description:

This is default overwrite setting for received files. Set to True (non-zero) to allow overwrites, else set to False (zero). This property only applies to ASCII and Xmodem downloads.

2.3.7.2.31 Device property

Syntax:

`session.Device`

Data type: integer

Description:

The communications device type attached to the session. The device type may be `atDevNONE` for no device (disconnects session), `atDevSERIAL`, `atDevTELNET`, `atDevSSH` or `atDevMODEM`.

2.3.7.2.32 DeviceLicenseMode property

Syntax:

`session.DeviceLicenseMode`

Data type: integer

Description:

If this setting is non-zero, "device licensing" negotiation between AccuTerm and the host system is enabled. When device licensing has been successfully negotiated, multiple AccuTerm connections (sessions) to the host system consume only one user license.

To enable device license negotiation, set this value to 1 if the device license response requires a *LF* termination character, or 3 if the response requires no termination character.

Device licensing is supported for UniVerse, UniData and QM host systems. The host system must have device licensing enabled (this is often an extra-charge item).

2.3.7.2.33 DisableAppMode property

Syntax:

session.DisableAppMode

Data type: boolean

Description:

Setting this property to True (non-zero) prevents AccuTerm from entering "keypad application mode" or "cursor application mode" when running one of the VT emulations. Set to False to allow application mode.

2.3.7.2.34 Duplex property

Syntax:

session.Duplex

Data type: integer

Description:

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).

2.3.7.2.35 ExtCols, ExtRows properties

Syntax:

session.ExtCols

session.ExtRows

Data type: integer

Description:

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

2.3.7.2.36 FKeys() property (settings)

Syntax:

settings.Fkeys(*index*)

Data type: string

Description:

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 special modifier for function key button bar caption can be combined with one of the standard modifiers to retrieve or store the caption & tooltip text. If both caption and tooltip are used, separate them with a pipe (|) character, caption first (a caption can be used without a tooltip). The string value of the programmed key is binary, and might not contain printable characters. See [session FKeys property](#) for modifier and virtual key values.

The FKeys property implements the standard VB Collection interface, and can be used in a For Each ... Next statement to enumerate all of the programmed keys for a session:

```
Sub Main()  
    Dim itm As Variant  
    Dim s As String  
    Dim key As String  
    Dim content As String  
    For Each itm In ActiveSession.Settings.FKeys  
        s = itm.Item  
        key = Item(s,1,1,vbNullChar)  
        content = Item(s,2,2,vbNullChar)  
        Debug.Print key & " = " & content  
    Next  
End Sub
```

Note: the Session FKeys property does not implement the Collection interface.

2.3.7.2.37 FontName property

Syntax:

session.FontName

Data type: string

Description:

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

2.3.7.2.38 FontSize property

Syntax:

session.FontSize

Data type: single

Description:

This is the size of the terminal font in points.

2.3.7.2.39 GmodeEnable property

Syntax:

session.GmodeEnable

Data type: boolean

Description:

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

2.3.7.2.40 Handshake property**Syntax:**

session.Handshake

Data type: integer

Description:

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` or `atDevMODEM`. Acceptable handshake settings are `atHandshakeNONE`, `atHandshakeXON`, `atHandshakeXIO`, `atHandshakeRTS` or `atHandshakeDTR`.

2.3.7.2.41 HistoryRows property**Syntax:**

session.HistoryRows

Data type: integer

Description:

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.

2.3.7.2.42 HostCertificateValidation property**Syntax:**

session.HostCertificateValidation

Data type: long

Description:

This property determines how the host certificate is validated when connecting using SSL/Telnet. Set this value to 0 if no validation is desired or -1 for strict validation. Otherwise, this is a bit mask with the following values:

- 1 = host certificate must be signed by a trusted certificate authority
- 2 = host certificate name must match host name
- 4 = host certificate must not be expired
- 8 = host certificate must not be revoked
- 16 = certificate must be valid for server authentication

2.3.7.2.43 HostKeyFingerprint property**Syntax:**

session.HostKeyFingerprint

Data type: string

Description:

This property contains the hash (fingerprint) of the SSH host key.

2.3.7.2.44 HostKeyValidation property

Syntax:

`session.HostKeyValidation`

Data type: long

Description:

When using an SSH connection, set this property to 0 to ignore the host's public key, 1 if the host key should be trusted if it has not yet been set, 2 to confirm any change in the host key, or 3 if the key must match the key fingerprint specified in the `HostKeyFingerprint` property.

2.3.7.2.45 HostName property

Syntax:

`session.HostName`

Data type: string

Description:

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 Telnet or Secure Shell. 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.

2.3.7.2.46 HostPort property

Syntax:

`session.HostPort`

Data type: string

Description:

For Telnet connections, this is the host TCP/IP port number assigned for Telnet services (default port is 23). For Secure Shell connections, it is the TCP/IP port assigned for SSH services (default port is 22).

2.3.7.2.47 HostTermType property

Syntax:

`session.HostTermType`

Data type: string

Description:

For Telnet and Secure Shell 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.

2.3.7.2.48 KeepaliveInterval property

Syntax:

`session.KeepaliveInterval`

Data type: long

Description:

If this setting is non-zero, AccuTerm will send a special "keepalive" message periodically to maintain the connection (some hosts and routers automatically disconnect when they detect an idle connection.)

This setting is in seconds. *Note: enabling keepalives when connecting to older versions of D3/NT may cause the error log on the host to fill up.)*

2.3.7.2.49 KermitCRC property

Syntax:

session.KermitCRC

Data type: integer

Description:

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.

2.3.7.2.50 KermitEOL property

Syntax:

session.KermitEOL

Data type: integer

Description:

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.

2.3.7.2.51 KermitErrRetry property

Syntax:

session.KermitErrRetry

Data type: integer

Description:

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

2.3.7.2.52 KermitInitRetry property

Syntax:

session.KermitInitRetry

Data type: integer

Description:

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

2.3.7.2.53 KermitQuote property

Syntax:

session.KermitQuote

Data type: integer

Description:

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.

2.3.7.2.54 KermitRept property

Syntax:

session.KermitRept

Data type: integer

Description:

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.

2.3.7.2.55 KermitTimeout property

Syntax:

session.KermitTimeout

Data type: integer

Description:

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

2.3.7.2.56 LegibleFont property

Syntax:

session.LegibleFont

Data type: boolean

Description:

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

2.3.7.2.57 LockBaudRate property

Syntax:

session.LockBaudRate

Data type: boolean

Description:

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.

2.3.7.2.58 LockFKeys property**Syntax:**

session.LockFKeys

Data type: integer

Description:

Set to 0 for unlocked function keys (host can reset or reprogram keys), 1 to lock from reset (host can reprogram keys, but not reset them) or 2 to lock from programming (host cannot reset or reprogram function keys).

2.3.7.2.59 MapUpperFKeys property**Syntax:**

session.MapUpperFKeys

Data type: boolean

Description:

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.

2.3.7.2.60 MessageRows property**Syntax:**

session.MessageRows

Data type: integer

Description:

This property sets the number of message rows and status lines for the session. The value must be between 0 and 3. Note that some emulations do not support any status or message rows, and some only support a single status line. Others support up to 3 lines. For more information, see [Status Lines](#).

2.3.7.2.61 MouseEnable property**Syntax:**

session.MouseEnable

Data type: boolean

Description:

Set to True (non-zero) to enable the ESC STX "1" / ESC STX "2" AccuTerm mouse-on command; otherwise set to False (zero).

2.3.7.2.62 NoAutoWrap property

Syntax:

session.NoAutoWrap

Data type: boolean

Description:

Set to True (non-zero) to disable automatic line wrap at end of line.

2.3.7.2.63 NormCols, NormRows properties

Syntax:

session.NormCols

session.NormRows

Data type: integer

Description:

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

2.3.7.2.64 OverrideModemConfig property

Syntax:

session.OverrideModemConfig

Data type: boolean

Description:

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).

2.3.7.2.65 Pages property

Syntax:

session.Pages

Data type: integer

Description:

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

2.3.7.2.66 Palette() property

Syntax:

session.Palette(index)

Data type: OLE_COLOR

Description:

This array contains the color values for the 16 palette entries used to display text on the terminal screen.

The [Color](#) property assigns foreground (text) and background colors based on these 16 colors. You can use the `RGB()` function to produce an `OLE_COLOR` from separate red, green and blue values. The red, green and blue values must be in the range of 0 to 255.

Palette entry	Color
0	Black
1	Dark Blue
2	Dark Green
3	Turquoise
4	Dark Red
5	Purple
6	Olive
7	Light Grey
8	Dark Grey
9	Blue
10	Green
11	Cyan
12	Red
13	Magenta
14	Yellow
15	White

An additional 6 palette entries exist for drawing borders (see [Visual Styles](#)). These are:

Palette entry	Used for
16	flat outer border
17	flat inner border
18	shadow outer border
19	shadow inner border
20	highlight outer border
21	highlight inner border

2.3.7.2.67 Parity property

Syntax:

`session`.Parity

Data type: integer

Description:

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 `atDevSERIAL` or `atDevMODEM`. Acceptable values are `atParityNONE`, `atParityEVEN`, `atParityODD`, `atParityMARK` and `atParitySPACE`. After changing this property, use the `Reset` method for the change to take effect.

2.3.7.2.68 PasteAllowCurlyQuotes property

Syntax:

session.PasteAllowCurlyQuotes

Data type: integer

Description:

Set to 0 (default) if AccuTerm should automatically translate "curly quotes" into "straight quotes" when pasting. Set to 1 to allow "curly quotes" to be pasted unchanged. *Note: "curly quotes" are only supported if your host character set encoding is "Unicode / UTF-8".*

2.3.7.2.69 PasteEOFChar property

Syntax:

session.PasteEOFChar

Data type: integer

Description:

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.

2.3.7.2.70 PasteEOFMode property

Syntax:

session.PasteEOFMode

Data type: integer

Description:

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.

2.3.7.2.71 PasteEOLChar property

Syntax:

session.PasteEOLChar

Data type: integer

Description:

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.

2.3.7.2.72 PasteEOLMode property

Syntax:

session.PasteEOLMode

Data type: integer

Description:

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. If you want to suppress the end-of-line character after the *last* line, add 128 to this value.

2.3.7.2.73 PasteSuppressFinalEOL property**Syntax:**

`session.PasteSuppressFinalEOL`

Data type: integer

Description:

Set to 1 if AccuTerm should omit the EOL character after the last line (or only line) when pasting. Set to 0 (default) to send EOL after each line.

2.3.7.2.74 Port property**Syntax:**

`session.Port`

Data type: integer

Description:

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 127. After changing this property, use the `Reset` method for the change to take effect.

2.3.7.2.75 PrinterColorMode property**Syntax:**

`session.PrinterColorMode`

Data type: integer

Description:

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).

2.3.7.2.76 PrinterFontBold property**Syntax:**

`session.PrinterFontBold`

Data type: boolean

Description:

Set this property to True (non-zero) to print using boldface. This property affects slave (or Aux) print jobs when the `SlavePrintMode` property is 1 (Graphics mode). This property has no effect when the `SlavePrintMode` property is zero (Text mode).

2.3.7.2.77 PrinterFontItalic property

Syntax:

`session.PrinterFontItalic`

Data type: boolean

Description:

Set this property to True (non-zero) to print using italics. This property affects slave (or Aux) print jobs when the `SlavePrintMode` property is 1 (Graphics mode). This property has no effect when the `SlavePrintMode` property is zero (Text mode).

2.3.7.2.78 PrinterFontName property

Syntax:

`session.PrinterFontName`

Data type: string

Description:

Set this property to the name of the font to use for slave (or Aux) print jobs when the `SlavePrintMode` property is 1 (Graphics mode). This property has no effect when the `SlavePrintMode` property is zero (Text mode).

2.3.7.2.79 PrinterFontSize property

Syntax:

`session.PrinterFontSize`

Data type: single

Description:

Set this property to the size (in points) of the font to use for slave (or Aux) print jobs when the `SlavePrintMode` property is 1 (Graphics mode). This property has no effect when the `SlavePrintMode` property is zero (Text mode).

2.3.7.2.80 PrinterOrientation property

Syntax:

`session.PrinterOrientation`

Data type: integer

Description:

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 1 (Graphics mode). This property has no effect when the `SlavePrintMode` property is zero (Text mode).

2.3.7.2.81 PrinterPaperSize property

Syntax:

session.PrinterPaperSize

Data type: integer

Description:

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 3.875 x 7.5 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 1/2 x 13 in	

2.3.7.2.82 PrinterPaperSource property

Syntax:

session.PrinterPaperSource

Data type: integer

Description:

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 1 (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

2.3.7.2.83 PrinterTimeout property

Syntax:

`session.PrinterTimeout`

Data type: integer

Description:

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

2.3.7.2.84 PrintJobEject property

Syntax:

`session.PrintJobEject`

Data type: integer

Description:

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.

2.3.7.2.85 PrintScreenBackground property

Syntax:

`session.PrintScreenBackground`

Data type: boolean

Description:

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

2.3.7.2.86 PrintScreenEject property**Syntax:**

session.PrintScreenEject

Data type: integer

Description:

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.

2.3.7.2.87 ProtectAttr property**Syntax:**

session.ProtectAttr

Data type: integer

Description:

Visual attribute number assigned to "protected fields" under Wyse/ADDs emulation. See [Colors\(\)](#) property for a list of attribute numbers.

2.3.7.2.88 ScaleFont property**Syntax:**

session.ScaleFont

Data type: boolean

Description:

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

2.3.7.2.89 ScreenPrintAdapterScript property**Syntax:**

session.ScreenPrintAdapterScript

Data type: string

Description:

This is the name of the print adapter script file to use for screen printing. Print adapter scripts are used to process the print job using an external program, such as Print Wizard. To use the print adapter script, the [ScreenPrintMode](#) property must be set to 2. See [Screen Printer Settings](#) for more information on print adapters.

2.3.7.2.90 ScreenPrinterName property**Syntax:**

session.ScreenPrinterName

Data type: string

Description:

This is the name of the printer used by the session for screen printing. It may be set to null ("") to indicate the current Windows default printer, or "no printer" to disable screen printing.

See also:

[SlavePrinterName property](#)

2.3.7.2.91 ScreenPrintMode property**Syntax:**

session.ScreenPrintMode

Data type: integer

Description:

Set to 0 for text mode, 1 for graphics mode or 2 to use the print adapter script specified by the [ScreenPrintAdapterScript](#) property. When this property is 1, the [PrinterColorMode](#) and [PrintScreenBackground](#) properties affect how the screen is printed.

2.3.7.2.92 ScreenPrintOrientation property**Syntax:**

session.ScreenPrintOrientation

Data type: integer

Description:

Set this property to 0 use the default page orientation, 1 for portrait or 2 for landscape. This property applies only to screen printing and only when the [ScreenPrintMode](#) property is 1 (Graphics mode). This property has no effect when the ScreenPrintMode property is zero (Text mode).

2.3.7.2.93 ScreenPrintPaperSize property**Syntax:**

session.ScreenPrintPaperSize

Data type: integer

Description:

Set this property to 0 use the default paper size for the printer. Otherwise use a paper size from the table shown in the [PrinterPaperSize](#) property. This property applies only when the [ScreenPrintMode](#) property is non-zero (Graphics mode). This property has no effect when the ScreenPrintMode property is zero (Text mode).

2.3.7.2.94 ScreenPrintPaperSource property**Syntax:**

session.ScreenPrintPaperSource

Data type: integer

Description:

Set this property to 0 use the default paper source for the printer. Otherwise use a paper source from the table shown in the [PrinterPaperSource](#) property. This property applies only to screen printing and only when the [ScreenPrintMode](#) property is 1 (Graphics mode). This property has no effect when the [ScreenPrintMode](#) property is zero (Text mode).

2.3.7.2.95 ScrMode property**Syntax:**

`session.ScrMode`

Data type: integer

Description:

Set to zero for normal (80 column) mode, 1 for extended (132 column) mode, or 2 for variable (programmers mode). *Note: when using variable mode, you must use a fixed size font.*

2.3.7.2.96 ScrollMode property**Syntax:**

`session.ScrollMode`

Data type: integer

Description:

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.

2.3.7.2.97 SlavePrintAdapterScript property**Syntax:**

`session.SlavePrintAdapterScript`

Data type: string

Description:

This is the name of the print adapter script file to use for slave (Aux) print jobs. Print adapter scripts are used to process the print job using an external program, such as Print Wizard. To use the print adapter script, the [SlavePrintMode](#) property must be set to 2. See [Slave Printer Settings](#) for more information on print adapters.

2.3.7.2.98 SlavePrinterName property**Syntax:**

`session.PrinterName`

`session.SlavePrinterName`

Data type: string

Description:

This is the name of the printer used by the session for slave (or Aux) printer output. It may be set to null

("") to indicate the current Windows default printer, or "no printer" to disable slave printing.

See also:

[ScreenPrinterName property](#)

2.3.7.2.99 SlavePrintMode property

Syntax:

session.SlavePrintMode

Data type: integer

Description:

Set to 0 for text mode, 1 for graphics mode or 2 to use the print adapter script specified in the [SlavePrintAdapterScript](#) property. When this property is 1, the *PrinterFont...*, *PrinterOrientation*, *PrinterPaperSize* and *PrinterPaperSource* properties affect how the slave (Aux) print job is printed. Print adapter scripts are used to process the print job using an external program, such as PrintWizard. See [Slave Printer Settings](#) for more information on print adapters.

2.3.7.2.100 Sound property

Syntax:

session.Sound

Data type: string

Description:

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 WAVE, MP3 or MIDI 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).

2.3.7.2.101 SSHAuth property

Syntax:

session.SSHAuth

Data type: integer

Description:

This property may be used to select the authentication method which will be used by a Secure Shell (ssh) session to authenticate a user to the host. Available authentication methods are: public key (2), password (3) and interactive (7). Use zero (0) for the default authentication method (password).

2.3.7.2.102 SSHCipher property

Syntax:

session.SSHCipher

Data type: integer

Description:

This property may be used to select a cipher to be used to encrypt a Secure Shell (ssh) session.

Available ciphers are: Triple DES (3), Blowfish (6), 128 bit AES (7) and 256 bit AES (8). Use zero (0) for the default cipher (Triple DES for SSH1 or 128 bit AES for SSH2).

2.3.7.2.103 SSHKey property**Syntax:**

session.SSHKey

Data type: string

Description:

This property contains the name of the SSH private key file. If no path is included, AccuTerm's default key folder will be used to read the key file.

2.3.7.2.104 SSHUseOnlyPreferredCipher property**Syntax:**

session.SSHUseOnlyPreferredCipher

Data type: boolean

Description:

Set to True (non-zero) to require that the cipher used for an SSH connection is the cipher specified by the [SSHCipher](#) property. If the host system does not accept the preferred cipher, the connection is aborted.

2.3.7.2.105 SSHVersion property**Syntax:**

session.SSHVersion

Data type: integer

Description:

This property may be used to select the desired SSH protocol version. Set this property to zero (0) for automatic protocol selection. Set to 1 for SSH version 1 or set to 2 for SSH version 2. When automatic protocol selection is used, AccuTerm selects the highest protocol version supported by the host.

2.3.7.2.106 StopBits property**Syntax:**

session.StopBits

Data type: integer

Description:

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 `atDevSERIAL` or `atDevMODEM`. Acceptable values are 1 or 2. After changing this property, use the `Reset` method for the change to take effect.

2.3.7.2.107 Strip8th property

Syntax:

session.Strip8th

Data type: boolean

Description:

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

2.3.7.2.108 TcpIdn property

Syntax:

session.TcpIdn

Data type: boolean

Description:

Set this property to True (non-zero) to enable Internationalized Domain Names (IDN) for the host name. IDN relies upon a standardized mechanism known as "Punycode" for encoding Unicode domain names using only the ASCII characters that are permitted by the global DNS system.

2.3.7.2.109 TcpNoDelay property

Syntax:

session.TcpNoDelay

Data type: boolean

Description:

Normally, the TCP network stack delays outbound network messages for a short time in order to "coalesce" multiple small messages into a single packet. When this setting is True (non-zero), AccuTerm will send each message as soon as possible. Setting this option may improve throughput, especially when running in AccuTerm's GUI environment and during file transfers, at the expense of increased network traffic.

2.3.7.2.110 TelnetBinary property

Syntax:

session.TelnetBinary

Data type: boolean

Description:

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

2.3.7.2.111 TelnetBypass property

Syntax:

session.TelnetBypass

Data type: boolean

Description:

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 D3 which do not implement the complete Telnet protocol.

2.3.7.2.112 TelnetKeepaliveIsNOP property

Syntax:

session.TelnetKeepaliveIsNOP

Data type: boolean

Description:

Set to True (non-zero) to send Telnet NOP command for keepalives; otherwise, sends Telnet DMARK command for keepalives.

2.3.7.2.113 TermType property

Syntax:

session.TermType

Data type: integer

Description:

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
atTermVT100	DEC VT-100
atTermVT220	DEC VT-220
atTermVT320	DEC VT-320
atTermVT420	DEC VT-420
atTermLinux	Linux console
atTermSCO	SCO console
atTermANSI	ANSI BBS
atTermPICKMON	Pick PC Monitor
atTermP30	MDIS P30
atTermTEK	Tektronix 4014
atTermTEK4100	Tektronix 4100

2.3.7.2.114 ThemeStyle property

Syntax:

session.ThemeStyle

Data type: string

Description:

This property contains the name of the current color theme, or "custom" if using customized colors.

2.3.7.2.115 XmodemTimeout property

Syntax:

session.XmodemTimeout

Data type: integer

Description:

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

2.3.7.2.116 YmodemTimeout property

Syntax:

session.YmodemTimeout

Data type: integer

Description:

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

2.3.7.2.117 ZmodemTimeout property

Syntax:

session.ZmodemTimeout

Data type: integer

Description:

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

2.3.8 AccuTerm constants

Description:

The following constants may be used in setting many of AccuTerm's properties.

Device constants

atDevNONE = 0	No communications device selected.
atDevSERIAL = 1	communications device is a serial port.
atDevPICLAN = 2	communications device is a PicLan network connection.

atDevTELNET = 3	communications device is a TCP/IP Telnet network connection.
atDevMODEM = 4	communications device is a modem.
atDevSSH = 5	communications device is a TCP/IP Secure Shell connection.

Serial port constants

atBaud300 = 0	Baud rate is 300 bps.
atBaud1200 = 1	Baud rate is 1200 bps.
atBaud2400 = 2	Baud rate is 2400 bps.
atBaud4800 = 3	Baud rate is 4800 bps.
atBaud9600 = 4	Baud rate is 9600 bps.
atBaud14400 = 5	Baud rate is 14,400 bps.
atBaud19200 = 6	Baud rate is 19,200 bps.
atBaud38400 = 7	Baud rate is 38,400 bps.
atBaud57600 = 8	Baud rate is 57,600 bps.
atBaud115200 = 9	Baud rate is 115,200 bps.
atParityNONE = 0	No parity checking
atParityODD = 1	Use odd parity
atParityEVEN = 2	Use even parity
atParityMARK = 3	Use mark parity
atParitySPACE = 4	Use space parity
atHandshakeNONE = 0	Do not use any flow control (handshake)
atHandshakeXON = 1	Use Xon/Xoff (software) flow control (inbound)
atHandshakeRTS = 2	Use RTS/CTS (hardware) flow control
atHandshakeDTR = 3	Use DTR/DSR (hardware) flow control
atHandshakeXIO = 4	Xon/Xoff handshake (bidirectional)

Duplex mode constants

atDuplexFULL = 0	Full duplex (host performs echo)
atDuplexHALF = 1	Half duplex (local echo)
atDuplexLOCAL = 2	Local mode

Terminal type constants

atTermTTY = 0	TTY emulation
atTermVPA2 = 1	ADDS Viewpoint A2 emulation
atTermVP60 = 2	ADDS Viewpoint 60 emulation
atTermP60 = 3	Procomm ADDS Viewpoint 60 emulation
atTermA2E = 4	ADDS Viewpoint A2 Enhanced emulation
atTermWY50 = 5	Wyse 50 emulation
atTermWY60 = 6	Wyse 60 emulation
atTermVT52 = 7	DEC VT52 emulation
atTermVT220 = 8	DEC VT220 emulation
atTermANSI = 9	ANSI BBS emulation
atTermPICKMON = 10	Pick PC Monitor (console) emulation

atTermTEK = 11	Tektronix 4014 emulation
atTermVT100 = 12	DEC VT100 emulation
atTermVT320 = 13	DEC VT320 emulation
atTermVT420 = 14	DEC VT420 emulation
atTermSCO = 15	SCO Console emulation
atTermLinux = 16	Linux Console emulation
atTermTEK4100 = 17	Tektronix 4100 emulation
atTermP30 = 18	Prism 30 emulation

File transfer constants

atProtocolASCII = 0	use ASCII file transfer protocol
atProtocolKermit = 1	use Kermit file transfer protocol
atProtocolXmodem = 2	use Xmodem file transfer protocol
atProtocolYmodem = 3	use Ymodem file transfer protocol
atProtocolZmodem = 4	use Zmodem file transfer protocol
atProtect = 0	do not overwrite existing file (Zmodem upload only)
atOverwrite = 1	overwrite existing file (Zmodem upload only)
atAppend = 2	append to existing file (Zmodem upload only)
atNewer = 3	overwrite existing file only when uploaded file is newer (Zmodem upload only)
atUpdate = 4	overwrite existing file, but ignore non-existing file (Zmodem upload only)
atResume = 5	Zmodem crash recovery

Reset constants

atResetAll = 0	Reset connection and terminal
atResetTerminal = 1	Reset terminal only
atResetComm = 2	Reset connection only

Printer constants

atPrinterOff = 0	Turn printer off
atPrinterOn = 1	Auto-print mode
atPrinterTransparent = 2	Transparent print mode

History scroll constants

atScrlHistLineUp = 0	Scroll history to previous line
atScrlHistLineDown = 1	Scroll history to next line
atScrlHistPageUp = 2	Scroll history to previous page
atScrlHistPageDown = 3	Scroll history to next page
atScrlHistTop = 6	Scroll history to beginning
atScrlHistBottom = 7	Scroll history to end

2.4 Mouse Support

Under normal conditions, clicking the left mouse button on the terminal screen will remove an active selection rectangle from the screen if there is one. Otherwise, a left-click is ignored. If you hold the left button down for about a half-second, you AccuTerm enters "selection mode" and you can drag the mouse to select an area of the screen. The selected area is enclosed with a dotted line.

If you click the terminal screen with the right mouse button, AccuTerm's context menu is shown. The context menu is a quick way to access common functions like copy & paste.

AccuTerm supports extended mouse functions which can be used to add mouse functionality to your character-based applications. Two mouse extensions are provided: *host mouse reporting*, and *mouse pattern matching*. The first requires modifications to the application program so that it can tell AccuTerm to send mouse clicks back to the host, as well as interpret the click information that AccuTerm sends back. The second uses a local pattern table. When AccuTerm recognizes a pattern under the mouse, it executes an action defined for the recognized pattern. Using mouse pattern matching generally does not require modifications to the host application.

Host mouse reporting

To use *host mouse reporting* you need to enable mouse support in [Mouse Settings](#). When the application wants AccuTerm to send mouse events, it needs to enable *mouse input* by sending a special escape sequence (see AccuTerm Programming - Mouse Functions). The mouse cursor changes to a pointing hand when *mouse input* is enabled. When it no longer wants to receive mouse events, it needs to disable *mouse input* by sending a different escape sequence. When *mouse input* is enabled, AccuTerm sends a report to the host whenever a mouse button is clicked on the terminal screen. The report indicates the character position of the click, the mouse button (left, right or center), and whether the a single-click or double-click has occurred.

Mouse pattern matching

To use *mouse pattern matching* you need to create a mouse pattern file, then tell AccuTerm to use the table by entering the file name in [Mouse Settings](#). Alternatively, the pattern table can be loaded dynamically using special escape sequences (see AccuTerm Programming - Mouse Functions).

The mouse pattern file is a tab delimited text file (ANSI character set). Each line in the file is an defines a pattern to match and the action for a click or double-click if a match is found. The patterns are *regular expressions* as are commonly used in many programming languages. Mouse pattern files are typically stored in the AccuTerm program directory (C:\Program Files\Atwin80).

Note: the Tab delimiters are important in this file. Using a word processor like Microsoft Word with formatting marks visible is recommended!

Each line in the mouse pattern file is made up of four fields separated by tabs:

button <tab> pattern <tab> click <tab> dblclk

The pattern file can contain up to 128 lines. Each line specifies the mouse ***button***, the ***pattern***, the action to take on a ***click*** and the action to take on 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 = center. Pattern and response strings are described below.

Patterns

The pattern field is a "regular expression", similar to the regular expressions used by the Linux `grep` command:

symbol	description
.	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

Actions

Action strings are encoded using the function key format (see [Keyboard Programming](#)). Just like function keys, if the action string is enclosed in square brackets ([]), AccuTerm treats the text between the brackets as a VBA script and executes it. If the action string is of the form [**name*], AccuTerm will display the context menu identified by *name*, which can be a custom context menu (see [Customizing the Menu and Toolbar](#)). For example, the action string [*reports] will display the popup menu named "reports". If the action string is not enclosed in brackets, it is transmitted to the host, after performing token substitutions as described below.

Tokens

Certain special tokens are permitted in the action string, whether it is enclosed in brackets (VBA script) or not. These tokens are useful for returning information to the host, or for passing as arguments to the script. These special tokens are:

%WORD%	replaced by the "word" which matched the pattern. The beginning and ending of "word" in the pattern are marked with the { and } symbols.
%COL%	replaced by the zero-based character column where the mouse was clicked.
%ROW%	replaced by the zero-based screen row where the mouse was clicked.

%BEG%	replaced by the zero-based character column where the matched word begins.
%END%	replaced by the zero-based character column where the matched word ends.
%MOVE%	replaced by the cursor movement commands required to position the cursor to the location of the mouse click. This is basically a string of up, down, left and right cursor position commands.
%%	replaced by the percent (%) character.

Examples

(todo)

3 AccuTerm 8 Reference

3.1 AccuTerm Programming

AccuTerm 8 supports many "private" commands which may be sent by host application programs. These commands allow for remote control of file transfer and data capture, executing DOS or Windows programs, enabling the mouse, displaying images, playing sounds, executing VBA scripts, and programming the function and keypad keys. These commands are valid in all terminal types.

External Program Execution

ESC STX < *command* CR

Executes the Windows command line ***command***, then returns to AccuTerm emulation mode immediately. Command is executed concurrently with the terminal session. ***Command*** may also be the file name of a document that is registered with Windows. If command is a document file name, the program associated with the specified file is used to open the file. For example, if the file name ends in ".doc", Microsoft Word is used to open the specified document file. You can also use this function to open a URL in the default browser (be sure to include the http:// prefix in the URL):

```
EQU ESC TO CHAR(27), STX TO CHAR(2), CR TO CHAR(13)
CMD = "calc.exe"
PRINT ESC:STX:'<':CMD:CR:
```

ESC STX > *command* CR

Executes the Windows command line ***command*** and suspends the terminal session until the command completes. See above for details.

Note: when using this function, you often need the host program to wait until the command terminates. If you require this functionality, then you should send the **ESC STX I** command immediately following the **ESC STX >** command, and then wait for input. AccuTerm will not process the **ESC STX I** until the command terminates. After the command completes, **ESC STX I** transmits AccuTerm's licensing information back to the host, thus the input is satisfied and the host program can continue execution:

```
EQU ESC TO CHAR(27), STX TO CHAR(2), CR TO CHAR(13)
CMD = "notepad.exe"
PRINT ESC:STX:'>':CMD:CR:
PRINT ESC:STX:'I':
INPUT JUNK
```

Mouse Functions

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. Both single and double clicks are supported. A single-click will always precede a double-click. *Note: this command is ignored unless host mouse reporting is*

enabled in the Mouse Settings.

The format of the mouse click report depends on whether the current terminal type uses ASCII or ANSI-style function keys. The report is designed to look like a function key, followed by the click position. For terminal types with ASCII-style function keys, the report is:

STX b CR ccc.rr CR

where **b** indicates which mouse button was pressed (**p**=left click, **q**=right click, **r**= center click, **P**=left double click, **Q**=right double click, **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).

For terminal types with ANSI-style function keys, the report is:

ESC [n ~ ESC [r ; c R

where **n** indicates which mouse button was pressed (101=left click, 102=right click, 103=center click, 111=left double click, 112=right double click, 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). If the terminal is using 8-bit control codes, AccuTerm sends **CSI** instead of **ESC [** in this report.

Note: if the mouse is clicked on an image that is displayed on the terminal screen, the location in the above reports is replaced by the image file name.

ESC STX 2

SystemBuilder compatibility mode; turns on mouse cursor. Transmits mouse location whenever a mouse button is pressed. *Note: this command is ignored unless host mouse reporting is enabled in the Mouse Settings.*

STX ~ CR b ; c ; r CR

b indicates which mouse button was pressed (1 = left, 2 = right, 4 = center), **c** is the column of the mouse cursor and **r** is the row of the mouse cursor (both in decimal, 0;0 is the upper left corner).

ESC STX h CR

Resets the mouse pattern table used for *mouse pattern matching*.

ESC STX h button HT pattern HT click HT dblclk CR

Adds an entry to the mouse pattern table used for *mouse pattern matching*. See Mouse Support for more information about using *mouse pattern matching*.

Data Transfer

ESC STX D p o m x ; 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**; transfer mode **m** may be **T** (text) or **B** (binary); translate mode **x**

may be **c** to enable character set translation or **q** to disable translation. **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.

Note: the download command is typically used by applications which implement the host side of the file transfer protocol to automatically start the corresponding client protocol. This command only affects the client (AccuTerm) side of the process.

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

Upload file from PC to host. Protocol **p** may be **A** (ASCII), **K** (Kermit), **X** (Xmodem), **Y** (Ymodem) or **Z** (Zmodem); transfer mode **m** may be **T** (text) or **B** (binary); translate mode **x** may be **c** to enable character set translation or **q** to disable translation. **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.

Note: the upload command is typically used by applications which implement the host side of the file transfer protocol to automatically start the corresponding client protocol. This command only affects the client (AccuTerm) side of the process.

ESC STX **S**

Returns status of last file transfer. The format of the 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
- 10 = destination file is write protected

ESC STX **C o p t ; 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. **T** may be **T** (text) to capture text only (filters out control characters) or null for no filtering. **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 Ƴ *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 file transfer settings. A *SUB* (CTRL+Z) is sent after the last line.

Status and Control**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 ?

Returns a string indicating the platform, product type, license type, capabilities and automation services. Message format is:

***platform * product * license * capabilities * services* CR**

where *platform* is 3 (Win32); *product* is 4 = AccuTerm (standard version), 5 = AccuTerm Internet Edition, 6 = AccuTerm Emulator ActiveX Control, 7 = AccuTerm Lite; release number, *serial* is the program serial number, *license* is 1 = single user, 2 = site, 3 = enterprise, 5 = developer, 7 = internet, 8 = component, 9 = evaluation; *capabilities* is a string of letters indicating the various capabilities where:

A = ASCII protocol supported
 B = Border effects (visual styles) enabled
 C = Capture supported
 D = Download supported
 E = Execute command supported
 F = On-the-fly character set conversion supported during file transfers
 G = GUI supported
 H = Server mode supported
 I = Image display supported
 J = Screen save & restore supported
 K = Kermit protocol supported
 L = Background image supported
 M = Query status supported
 O = Object Bridge supported
 P = Packetized messages supported
 Q = Message integrity checks supported
 R = Reliable connection
 S = Scripting supported
 T = File transfer error info supported
 U = Upload supported
 V = Host capabilities command supported
 W = Synchronous scripts supported

X = File conversion supported

Y = Printer control & status query supported

services is a string of letters indicating which automation services are available (**a** = Object Bridge, **b** = file converter, **g** = GUI).

ESC STX = *host-capabilities* * *buffer-size* CR

Informs AccuTerm of any special host program capabilities. ***host-capabilities*** is a string of letters indicating various host features:

D = host programs use tag notation for system delimiters

C = host supports message checksums

M = host supports message length check

buffer-size is optional, and specifies the maximum packet size that the client can send to host.

ESC STX % 0

Returns the program directory, followed by **CR**. This is normally the directory where AccuTerm is installed, except in the case of AccuTerm Internet Edition, the directory returned is the one containing the Internet Explorer exe file.

ESC STX % 1

Returns the current directory (usually from the shortcut "start in" field), followed by **CR**.

ESC STX % 2

Returns the directory containing AccuTerm's atwin71.ini file, followed by **CR**.

ESC STX % 3

Returns the directory containing AccuTerm's emulator DLL, followed by **CR**.

ESC STX x

Terminates the current 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 L

Sets the CapsLock key state to "On"

ESC STX M

Sets the CapsLock key state to "Off"

Terminal Screen

ESC STX E

Selects extended (132 column) screen size as defined by the extended columns and rows in the screen settings.

ESC STX N

Selects normal (80 column) screen size as defined by the normal columns and rows in the screen settings.

ESC STX A color CR

Sets the foreground (text) color to **color**. This command overrides the foreground color assigned by the last visual attribute selected.

ESC STX B color CR

Sets the background color to **color**. This command overrides the background color assigned by the last visual attribute selected.

Color	Display
0	Black
1	Dark Blue
2	Dark Green
3	Turquoise
4	Dark Red
5	Purple
6	Olive
7	Light Gray
8	Dark Grey
9	Blue
10	Green
11	Cyan
12	Red
13	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 (see Color Settings).

ESC STX r x1 , y1 , x2 , y2 , fill , border CR

Draw rectangle. **x1, y1** is the upper-left corner and **x2, y2** is the lower-right corner. Coordinates are in character columns and rows. The upper-left corner of the screen is (0,0).

The rectangle can be filled using either a visual attribute or a background color. To fill with a visual attribute, **fill** is **A0 .. A63**. The attribute number is formed by adding the following values:

0 = normal
 1 = invisible
 2 = blinking
 4 = reverse
 8 = underline
 16 = dim
 32 = bright

To fill with a background color (as described above), **fill** is **B0 ... B15**. To draw a border without filling the rectangle, **fill** is **r**. A border can be drawn around the rectangle. **Border** is **r** (no border), **B** (flat border), **I** (inset border) or **R** (raised border).

ESC STX js , ID , col , row , width , height , page CR

Saves a copy of the specified screen block (text, colors and visual effects) and terminal state in memory and associates the block with the specified identifier (**ID**) which can be an arbitrary alpha-numeric string (may not contain commas). Any number of screen blocks can be saved. **Col** and **row** specify the upper-left corner of the block. If either is omitted or null, zero is assumed. **Width** and **height** specify the size of the block. If **width** or **height** is omitted or null, the screen width or height is used. **Page** optionally specifies the terminal page for the block, and if omitted or null, the current page is used.

ESC STX jR , ID , col , row , page , state CR

Restores the screen block associated with the specified identifier (**ID**) to the screen. **Col** and **row** specify the upper-left corner of the destination. If either is omitted or null, the original position of the saved block is assumed. **Page** optionally specifies the terminal page for the destination, and if omitted or null, the current page is used. **State** optionally specifies whether the terminal state (cursor position, visual attribute, text color, etc.) is restored. To restore the terminal state, pass 1 for this argument.

ESC STX jD , ID CR

Deletes the screen block associated with the specified identifier (**ID**) from memory.

ESC STX yj , ID CR

Queries status of a stored screen block. Sends a 0 or 1, followed by a **CR**, indicating if a block of the specified identifier (**ID**) exists in AccuTerm's screen block memory.

Keyboard**ESC STX F t s d 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 **x** (keypad keys). Both **c** and **x** are permitted when programming control-keypad keys. Shift **s** may be **u** (unshifted) or **s** (shifted). Destination **d** may be null (program key) or **T** (program function bar button caption). Key code **k** may be digits 0 to 9 or : ; < = > according to the following table. **Data** contains the function key data (or caption text). Normally, the key data is transmitted to the host when the programmed key is pressed. However, if **data** is enclosed in brackets ([]), then when the key is pressed, the key data will be interpreted as a VBA script statement and will be executed instead of being sent to the host. When setting both the caption and the tooltip text for a function bar button, **data** is the caption text followed by a pipe (|) character, followed by the tooltip text.

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

To enter control characters (and special prefix characters) in key data, use the notation shown in the following table:

Key sequence	Control code	AccuTerm notation	Key sequence	Control code	AccuTerm notation
CTRL + @	NUL	^@	CTRL + P	DLE	^P
CTRL + A	SOH	^A	CTRL + Q	DC1	^Q
CTRL + B	STX	^B	CTRL + R	DC2	^R
CTRL + C	ETX	^C	CTRL + S	DC3	^S
CTRL + D	EOT	^D	CTRL + T	DC4	^T
CTRL + E	ENQ	^E	CTRL + U	NAK	^U
CTRL + F	ACK	^F	CTRL + V	SYN	^V
CTRL + G	BEL	^G	CTRL + W	ETB	^W
CTRL + H	BS	^H	CTRL + X	CAN	^X
CTRL + I	HT	^I	CTRL + Y	EM	^Y
CTRL + J	LF	^J	CTRL + Z	SUB	^Z
CTRL + K	VT	^K	CTRL + [ESC	^[
CTRL + L	FF	^L	CTRL + \	FS	^\
CTRL + M	CR	^M	CTRL +]	GS	^]
CTRL + N	SO	^N	CTRL + ^	RS	^~
CTRL + O	SI	^O	CTRL + _	US	^_
^		^^	\		\\
DELETE	DEL	^?			

For example, to program the END key to send the word "END", followed by a carriage return, the following code could be used:

```
EQU ESC TO CHAR(27), STX TO CHAR(2), CR TO CHAR(13)
PRINT ESC:STX:'FKU5END^M':CR:
```

Printer

ESC STX p mode CR

Selects the printer mode:

mode = 0 - turns off auto or transparent print

mode = 1 - turns on auto (copy) print mode

mode = 2 - turns on transparent print mode

mode = X - turns off auto or transparent print and closes the print job immediately

Multimedia

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 BMP, JPEG, GIF, PCD, PCX, PICT, PNG, PSD, TARGA, TIFF, WBMP, XBM, XPM and Windows Metafile. Images are considered "protected" data; a "clear unprotected characters" command will not clear images from the terminal screen.

The image file does not need to be a local file; if an internet connection is available the filename can specify a URL instead of a local or network file.

Note: AccuTerm 8 uses the FreeImage Open Source image library.

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 iB, filename, scalemode, opacity CR

Loads an image to use as the screen background (wallpaper).

filename - the path or URL of the image to use as the screen background.

scalemode = 0 - picture is stretched to fit screen size. This may alter the picture's aspect ratio.

scalemode = 1 - picture is scaled so that its longest dimension fills the screen. The area not filled by the picture is black.

scalemode = 2 - picture is scaled so that its shortest dimension fill the screen. The longer dimension is cropped.

opacity - specifies the opacity of the background image, as a percentage. 100 means that the picture is opaque. 0 means that the image is invisible (transparent) and the screen background color is used as the background. Any percentage in between blends the image with the screen background color.

To clear the background image, simply use **ESC STX iB CR**.

ESC STX m filename CR

ESC STX w filename CR

Plays sound file **filename**. The file does not need to be a local file; if an internet connection is available the filename can specify a URL instead of a local or network file. Supported sound file formats are WAVE, MIDI and MP3.

Scripting

ESC STX P statements CR

Executes VBA script **statements**. See Scripting for details on VBA scripting. Multiple statements may be executed; separate each statement with **LF** or **EM** control characters. You can call subroutines or functions contained within the script loaded in the Main Script tab of the Script Editor window. For example, to call subroutine **foo**, simply send: **ESC STX P foo CR**. You can execute a script stored in a script file (.atsc file) by using the Chain statement:

```
EQU ESC TO CHAR(27), STX TO CHAR(2), CR TO CHAR(13)
FILENAME = 'myscriptfile.atasc'
SCRIPT = 'Chain "':FILENAME:''
PRINT ESC:STX:'P':SCRIPT:CR:
```

ESC STX R *statements* CR

Executes VBA script ***statements*** (see above) except that the script is synchronized with the terminal state. This basically sets the session's `InputMode` property to 1 before the script starts and restores it once the script completes. This ensures that any characters received from the host will not be processed by the terminal emulator unless they are first processed in the script using `Input` or `WaitFor` functions. This has the effect of buffering any data, control codes or escape sequences received while the script is running. This is useful when changing `Session` properties such as screen sizes, terminal type, etc.

*Note: in previous versions of AccuTerm, synchronization required adding a final `Output` method in the script and corresponding `INPUT` statement in the host program. However, this technique is not compatible with typeahead on the host and is less efficient than **ESC STX R** and is no longer recommended.*

3.2 Wyse Programming

This topic 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 values and their function are shown in the [Wyse Operating Mode Table](#).

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 **S0**

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.

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 message 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 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.

ESC e &

Set **CapsLock** keyboard state.

ESC e '

Clear **CapsLock** keyboard state.

ESC e .

Do not clear screen when screen size changes.

ESC e /

Clear screen when screen size changes.

ESC ^ 0

Normal display (light characters, dark background).

ESC ^ 1

Reverse display (dark characters, light background).

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 ~ 8

Select Viewpoint 60 emulation.

ESC ~ ;

Select VT-100 emulation.

ESC ~ <

Select VT-220 7-bit emulation.

ESC ~ = or **ESC ~ A**

Select VT-220 8-bit emulation.

ESC ~ B

Select VT-320 7-bit emulation.

ESC ~ C

Select VT-320 8-bit emulation.

ESC ~ >

Select Tektronix 4014 emulation.

ESC c v or **ESC c w**

Save current session settings to disk. If no session file name exists, then one will be prompted for.

ESC c x

Reset (reload session configuration file).

Character Set Selection

Wyse terminals have two character sets (primary and secondary) of 128 symbols each, and four font banks of 128 symbols. Each font bank can be assigned a pre-defined symbol set (like PC Multinational), and each character set can be assigned to one font bank.

By default, the native Wyse 50 symbol set is assigned to font banks 0, 2 and 3. The PC Multinational symbol set is assigned to font bank 1. Font bank 0 is assigned to the primary character set and font bank 1 is assigned to the secondary character set.

ESC c D

Selects the primary character set. This is the default, and when selected, the primary character set is displayed for character codes 0 to 127, and the secondary character set is displayed for codes 128 to 255.

ESC c E

Selects the secondary character set. When the secondary character set is selected, it is displayed for character codes 0 to 127 and again for codes 128 to 255.

ESC c B *bank*

Assigns font bank ***bank*** (0, 1, 2 or 3) to the primary character set.

ESC c C *bank*

Assigns font bank ***bank*** (0, 1, 2 or 3) to the secondary character set.

ESC c @ *bank set*

Assigns pre-defined symbol set ***set*** to font bank ***bank***. ***Set*** can be one of the following:

<i>set</i>	<i>symbol set</i>
@ or ~	Native Wyse 50

set	symbol set
A Or a	PC Multinational
B , c , d , G , H Or g	US ASCII
J , j , N , I Or e	DEC Graphics / ISO Latin-1
D or b	PC Standard

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.

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).

SOH

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](#).[‡]

DLE *c*

Moves the cursor to column *c* where *c* is a valid column in the from the [Viewpoint Cursor Address Table](#).[‡]

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](#). 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](#). 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 _ *c*

Moves the cursor to column *c* where *c* is a single byte column address from the [Wyse Cursor Address Table](#). Note that this command cannot position the cursor past column 95.[‡]

ESC [*r*

Moves the cursor to row *r* where *r* is a single byte row address from the [Wyse Cursor Address Table](#).

ESC]

Selects page 0.

ESC }

Selects page 1.

ESC - *n r c*

Selects page *n* (pages numbered from 0) and positions the cursor to row *r*, column *c*. *r* and *c* are single byte cursor address codes from the [Wyse Cursor Address Table](#). To select pages higher than 9, use CHAR(*n* + 48).

ESC w n

Selects page *n*. Pages are numbered from 0 to 9.

ESC w C

Page forward. If currently on the last page, the first page is selected.

ESC K

Page forward. If currently on the last page, the first page is selected.[†]

ESC w B

Page back. If currently on the first page, the last available page is selected.

ESC J

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) and positions the cursor to row *r*, column *c*. *r* and *c* are single byte cursor address codes from the [Wyse Cursor Address Table](#). To select pages higher than 9, use CHAR(*n* + 48).

ESC x A r

Splits screen horizontally at row *r*, where *r* is a valid row code from the [Wyse Cursor Address Table](#) or from the [Viewpoint Cursor Address Table](#) if in Viewpoint Enhanced mode.

ESC x I r

Splits screen horizontally at row *r* and clear screen, where *r* is a valid row code from the [Wyse Cursor Address Table](#) or from the [Viewpoint Cursor Address Table](#) if in Viewpoint Enhanced 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). If the terminal is in "protect mode", only unprotected characters are cleared.[†]

ESC ;

Clears the screen and returns the cursor to the home position (upper left corner of the screen). If the terminal is in "protect mode", only unprotected characters are cleared.

ESC * or ESC +

Clears the screen and move the cursor to the home position (upper left corner of the screen). Both protected and unprotected characters are cleared. Turns off the protect mode.

ESC T or ESC t

Clears from the cursor position to the end of the current line. If the terminal is in "protect mode", only unprotected characters are cleared.

ESC K

Clears from the cursor position to the end of the current line.[‡]

ESC Y

Clears from the cursor position to the end of the screen. If the terminal is in "protect mode", only unprotected characters are cleared.[†]

ESC y

Clears from the cursor position to the end of the screen. If the terminal is in "protect mode", only unprotected characters are cleared.

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).

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. Columns and rows are numbered from 1, pages are numbered from 0.

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.

AccuTerm's Viewpoint Enhanced emulation supports most ADDS 4000 functions, including support for both non-embedded attributes like Wyse 60 and embedded attributes like Wyse 50.

ESC 0 a

Assigns the "tagged" video attribute. *a* is the video attribute code from the [Viewpoint Attribute Code Table](#).[‡]

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](#). The application display field codes are:

Field (<i>n</i>)	Display area
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. In Wyse 50 and ADDS VP mode, this command displays a space at the cursor position, then changes the visual attribute for all characters from this position until the end of the screen or until another attribute code 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](#).

ESC g a

Assign visual attribute. 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](#). ‡

ESC g n or **ESC g n** ‡

Assign line attribute. This is the same command as is used to assign visual attributes, but with different attribute codes. This sequence is used to specify character size for the current line. The line attribute code **n** is:

n	attribute
@	normal size characters
A	double-wide characters
B	double-high characters, top
C	double-high characters, bottom
D	double-wide, double-high characters, top
E	double-wide, double-high characters, bottom

ESC HT n

Assign line attribute (performs the same function as above using different attribute codes): ‡

n	attribute
0	normal size characters
1	double-wide characters
2	double-wide, double-high characters, top
3	double-wide, double-high characters, bottom

Protected Attributes

Protected attributes are used by some software applications 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 non-protected characters can be cleared by using the "clear unprotected" 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 .
ESC . c	Replaces all unprotected characters with the character c .
ESC ; or ESC : or SUB	Clears all unprotected characters.
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.
ESC cR or ESC cS	Clears unprotected characters from the cursor position to the end of the current line. Attributes are not cleared.
ESC cP or ESC cQ	Clears unprotected characters from the cursor position to the end of the screen. Attributes are not cleared.

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 . 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](#).

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 **CR**.

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 4

Sends unprotected characters in the current row to the host computer, from the beginning of the row up to the cursor position. The row is terminated by a **CR**.[†]

ESC 5

Sends all unprotected characters on the screen to the host computer, beginning with the upper-left corner, and ending at the cursor position. Each row except the last is terminated by a **US** control code. The last row is terminated by a **CR**.[†]

ESC 6

Sends the current row to the host computer, from the beginning of the row, up to the cursor position. The row is terminated by a **CR**.[†]

ESC 7

Sends the screen to the host computer, beginning with the upper-left corner, and ending at the cursor position. Each row except the last is terminated by a **US** control code. The last row is terminated by a **CR**.

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 address to the host computer in 132 column format. No **CR** is sent

after the address.

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".

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).

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](#).

ESC z f label CR

Program function key label field *f* as *label*. The field codes are specified in the [Wyse Function Key Table](#).

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](#).

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 _ f label EM

Program function key label field *f* as *label*. The function key field codes are specified in the [Viewpoint Function Key Table](#). ‡

ESC f text CR

Sets the unshifted label line to *text*. If *text* is omitted, clears the unshifted label line.

ESC % k 0n sequence EM

Program function or keypad key *k* to send *sequence* to host when pressed. The function and keypad key codes are specified in the [Viewpoint Function Key Table](#). ‡

ESC c U

Clear all redefinable key definitions to their default values.

ESC z ~ k

Read programmable key definition for key *k*.

3.2.1 Wyse Tables

3.2.1.1 Wyse Operating Mode Table

The **Wyse Operating Mode Table** defines the operating mode (*n*) used by the Wyse emulations to set the terminal operating mode.

<i>n</i>	Function
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

<i>n</i>	Function
<i>;</i>	132 column screen
B	Protect blink on
C	Protect invisible on
E	Protect underline on
F	Protect reverse on
G	Protect dim on

3.2.1.2 Wyse Cursor Address Table

The **Wyse Cursor Address Table** shows the character code used by the Wyse emulations to specify cursor column (*c*) or row (*r*).

Row (<i>r</i>) or column (<i>c</i>)	Code	Row (<i>r</i>) or column (<i>c</i>)	Code	Row (<i>r</i>) or column (<i>c</i>)	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	~

Row (r) or column (c)	Code	Row (r) or column (c)	Code	Row (r) or column (c)	Code
32	?	64	—	96	DEL

3.2.1.3 Wyse Attribute Code Table

The **Wyse Attribute Code Table** defines the attribute code character (**a**) used by the Wyse emulations to assign visual attributes.

Code (a)	Attribute Type
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, blink, 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
DEL	Dim, underline, reverse, blink, blank

3.2.1.4 Wyse Function Key Table

The **Wyse Function Key Table** defines the key code character (**k**) used by the Wyse emulations to program function and keypad keys. Also defines the field code character (**f**) used for programming

function key labels.

Key	Normal key code (k)	Shifted key code (k)	Normal field (f)	Shifted field (f)
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		

3.2.1.5 Wyse Key Code Table

The **Wyse Key Code Table** describes the default character sequence transmitted by the function, keypad and other special keys on the PC keyboard and the corresponding real terminal key.

















PC Key	Terminal Key	Sequence
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>

PC Key	Terminal Key	Sequence
INS	INS	ESC q
SHIFT+INS	SHIFT+INS	ESC r
DEL	DEL	ESC w
SHIFT+DEL	DEL LINE	ESC R
HOME	HOME	RS
SHIFT+HOME	SHIFT+HOME	ESC {
END	CLR LINE	ESC T
SHIFT+END	CLR SCRN	ESC Y
PGUP	PGUP	ESC J
SHIFT+PGUP	SHIFT+PGUP	ESC J
PGDN	PGDN	ESC K
SHIFT+PGDN	SHIFT+PGDN	ESC K
LEFT	LEFT	BS
SHIFT+LEFT	SHIFT+LEFT	BS
RIGHT	RIGHT	FF
SHIFT+RIGHT	SHIFT+RIGHT	FF
UP	UP	VT
SHIFT+UP	SHIFT+UP	VT
DOWN	DOWN	LF
SHIFT+DOWN	SHIFT+DOWN	LF
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
F1	F1	SOH @ CR
SHIFT+F1	SHIFT+F1	SOH ` CR
F2	F2	SOH A CR
SHIFT+F2	SHIFT+F2	SOH a CR
F3	F3	SOH B CR
SHIFT+F3	SHIFT+F3	SOH b CR
F4	F4	SOH C CR
SHIFT+F4	SHIFT+F4	SOH c CR
F5	F5	SOH D CR
SHIFT+F5	SHIFT+F5	SOH d CR
F6	F6	SOH E CR
SHIFT+F6	SHIFT+F6	SOH e CR
F7	F7	SOH F CR
SHIFT+F7	SHIFT+F7	SOH f CR
F8	F8	SOH G CR
SHIFT+F8	SHIFT+F8	SOH g CR
F9	F9	SOH H CR
SHIFT+F9	SHIFT+F9	SOH h CR
F10	F10	SOH I CR
SHIFT+F10	SHIFT+F10	SOH i CR

PC Key	Terminal Key	Sequence
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>

3.2.1.6 Wyse Graphic Character Table

The **Wyse Graphic Character Table** defines the character code used for each of the line-drawing graphic characters used by the Wyse emulations.

Graphic	Code	Graphic	Code
	0		8
	1		9
	2		:
	3		;
	4		<
	5		=
	6		>
	7		?

3.2.2 Wyse Programming Notes

† Functions tagged with the single-dagger symbol are only valid in Wyse 50 or Wyse 60 emulation.

‡ Functions tagged with the double-dagger symbol are only valid in Viewpoint Enhanced emulation.

3.3 ADDS Programming

This topic 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.

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.

SOH

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](#).

DLE *c*

Moves the cursor to column *c* where *c* is a valid column in the from the [Viewpoint Cursor Address Table](#).

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](#). 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 *l*

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. ◇

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. Only one attribute may be displayed on the screen at one time - all characters which are "tagged" are displayed using the current tag attribute. *Other emulations, including Viewpoint A2 Enhanced and ADDS 4000, allow multiple attributes to be displayed simultaneously.*

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.

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](#).

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](#).

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](#).

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](#).
◇

ESC 2

Turn off line graphics mode. ◇

Printer Control

DC2

Auto (copy) print mode. Characters are displayed and printed. This command will be ignored if the current slave printer selection is "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).

3.3.1 Viewpoint Tables

3.3.1.1 Viewpoint Cursor Address Table

The **Viewpoint Cursor Address Table** shows the character code used by the Viewpoint emulations to specify cursor column (c) or row (r).

Column (c)	Code	Column (c)	Code	Column (c)	Code
1	<i>NUL</i>	36	5	71	<i>p</i>
2	<i>SOH</i>	37	6	72	<i>q</i>
3	<i>STX</i>	38	7	73	<i>r</i>
4	<i>ETX</i>	39	8	74	<i>s</i>
5	<i>EOT</i>	40	9	75	<i>t</i>

Column (c)	Code	Column (c)	Code	Column (c)	Code
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	Row (r)	Code
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

3.3.1.2 Viewpoint Attribute Code Table

The **Viewpoint Attribute Code Table** defines the attribute code character (**a**) used by the Viewpoint emulations to assign visual attributes.

Code (a)	Attribute Type
@	Normal
A	Dim
B	Normal Blinking
C	Dim Blinking
P	Reverse
Q	Dim Reverse
R	Reverse Blinking
S	Dim Reverse Blinking
`	Underlined

Code (<i>a</i>)	Attribute Type
a	Dim Underlined
b	Underlined Blinking
c	Dim Underline Blinking
D	Invisible

3.3.1.3 Viewpoint Function Key Table

The **Viewpoint Function Key Table** defines the key code character (*k*) used by the Viewpoint emulations to program function and keypad keys.

Key	Normal key code (<i>k</i>)	Shifted key code (<i>k</i>)
F1	<i>NUL</i>	<i>DLE</i>
F2	<i>SOH</i>	<i>DC1</i>
F3	<i>STX</i>	<i>DC2</i>
F4	<i>ETX</i>	<i>DC3</i>
F5	<i>EOT</i>	<i>DC4</i>
F6	<i>ENQ</i>	<i>NAK</i>
F7	<i>ACK</i>	<i>SYN</i>
F8	<i>BEL</i>	<i>ETB</i>
F9	<i>BS</i>	<i>CAN</i>
F10	<i>HT</i>	<i>EM</i>
F11	<i>LF</i>	<i>SUB</i>
F12	<i>VT</i>	<i>ESC</i>
F13	<i>FF</i>	<i>FS</i>
F14	<i>CR</i>	<i>GS</i>
F15	<i>SO</i>	<i>RS</i>
F16	<i>SI</i>	<i>US</i>
BKSP	7	W
TAB	8	X
INS	;	[
HOME	<i>SPACE</i>	@
END	#	C
PGUP	!	A
PGDN	"	B
LEFT	%	E
RIGHT	&	F
UP	'	G
DOWN	\$	D
ESC	6	V
ENTER	9	Y
KPD ENTER	5	U

3.3.1.4 Viewpoint Key Code Table

The **Viewpoint Key Code Table** describes the default character sequence transmitted by the function, keypad and other special keys on the PC keyboard and the corresponding real terminal key.

PC Key	Terminal Key	Sequence
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>ESC w</i>
SHIFT+DEL	DEL LINE	<i>ESC l</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>
F1	F1	<i>STX 1 CR</i>
SHIFT+F1	SHIFT+F1	<i>STX ! CR</i>
F2	F2	<i>STX 2 CR</i>
SHIFT+F2	SHIFT+F2	<i>STX " CR</i>
F3	F3	<i>STX 3 CR</i>
SHIFT+F3	SHIFT+F3	<i>STX # CR</i>
F4	F4	<i>STX 4 CR</i>
SHIFT+F4	SHIFT+F4	<i>STX \$ CR</i>
F5	F5	<i>STX 5 CR</i>
SHIFT+F5	SHIFT+F5	<i>STX % CR</i>
F6	F6	<i>STX 6 CR</i>
SHIFT+F6	SHIFT+F6	<i>STX & CR</i>
F7	F7	<i>STX 7 CR</i>
SHIFT+F7	SHIFT+F7	<i>STX ' CR</i>
F8	F8	<i>STX 8 CR</i>
SHIFT+F8	SHIFT+F8	<i>STX (CR</i>
F9	F9	<i>STX 9 CR</i>

PC Key	Terminal Key	Sequence
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

3.3.1.5 Viewpoint 60 Graphic Character Table

The **Viewpoint 60 Graphic Character Table** defines the character code used for each of the line-drawing graphic characters used by Viewpoint 60 emulation.

Graphic	Code
┌	@ 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

3.3.2 ADDS Programming Notes

◇ Functions tagged with the diamond symbol are only valid in Viewpoint 60 or Procomm VP60 emulation.

3.4 ANSI Programming

The AccuTerm ANSI emulations provide **VT52**, **VT100**, **VT220**, **VT320**, **VT420**, **SCO Console**, **Linux Console** and **ANSI BBS** emulations. The VT220, VT320 and VT420 emulations support international characters, 7 and 8 bit control codes, programmable function keys and a robust command set. VT100 supports 7 bit control codes and a subset of the VT220 command set.

The SCO Console emulation uses the PC (OEM) character set and supports programmable function keys and text and background colors.

The Linux Console emulation uses the ANSI character set, and includes support for programmable function keys, text and background colors, and mouse.

The ANSI BBS emulation supports the PC (OEM) 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 BBS emulation only supports 7 bit escape sequences.

When AccuTerm is emulating a VT220, VT320 or VT420 terminal snipor SCO or Linux Console, it will respond to 7-bit and 8-bit control codes (hex 00-1F and 80-9F). For convenience, 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 [
SS3	=	ESC O
DCS	=	ESC P
ST	=	ESC \

This chapter documents commands used by all of the ANSI emulations supported by AccuTerm. Not every command is valid for all ANSI emulations. Those commands which are only valid for certain emulations are shown with a reference note after the command definition. These reference are indicated by a superscript, and are VT100 ¹, VT220 ², VT320 ³, VT420 ⁴, ANSI BBS ^A, Linux Console ^L, and SCO Console ^S.

The AccuTerm ANSI emulations operate in a variety of modes. Some of the default operating modes are determined by settings in the AccuTerm configuration. Most of these modes can be changed by commands received from the host. Operating modes which can be selected in the Settings dialog and saved in a configuration file are:

- VT52, VT100, VT220, VT320, VT420, SCO Console, Linux Console or ANSI BBS emulation
- 80 (normal) or 132 (extended) columns
- Automatic wrap at end of line
- Send 7 or 8 bit control codes
- Numeric keypad sends “application” codes instead of numbers
- Cursor keys send “application” codes instead of cursor codes

- Backspace key sends *DEL* control code instead of *BS* control code

The following operating modes are not determined by settings in the AccuTerm configuration, but may be changed by sending the appropriate ANSI command sequence:

- Keyboard: default is unlocked.
- Insert/Replace: default is replace.
- Line feed / New line: default is line feed.
- Origin: default is absolute.
- 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 VT220, VT320 and VT420 emulations support 5 character sets: ASCII, Multinational, ISO-Latin1, Graphics and Scientific. The default character set assignment is G0=ASCII, G1=ASCII, G2=Multinational and G3=ASCII. The GL set (hex 20-7E) defaults to G0 (ASCII) and the GR set (hex A0-FE) defaults to G2 (Multinational).

Operating Modes

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.²³⁴

CSI n + p

Secure 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. If *n* is non-zero, then AccuTerm responds by sending **CSI n * q** back to the host. The value of *n* must be between 0 and 16383.¹²³⁴

ESC SPACE F

Causes AccuTerm to send 7-bit control codes:²³⁴

CSI	=	ESC [
SS3	=	ESC O
DCS	=	ESC P
ST	=	ESC \

ESC SPACE G

Causes AccuTerm to send 8-bit control codes *CSI* , *SS3*, *DCS* and *ST*.²³⁴

CSI 6 1 " p

Changes the emulation to VT100.²³⁴

CSI 6 2 ; n " p

Changes the emulation to VT220. If *n* = 1, AccuTerm will equivalent send 7-bit escape sequences for control codes *CSI* , *SS3*, *DCS* and *ST*; otherwise, 8-bit control codes will be sent.²³⁴

CSI 6 3 ; n " p

Changes the emulation to VT320. If *n* = 1, AccuTerm will equivalent send 7-bit escape sequences for control codes *CSI* , *SS3*, *DCS* and *ST*; otherwise, 8-bit control codes will be sent.²³⁴

CSI 6 4 ; n " p

Changes the emulation to VT420. If *n* = 1, AccuTerm will equivalent send 7-bit escape sequences for control codes *CSI* , *SS3*, *DCS* and *ST*; otherwise, 8-bit control codes will be sent.²³⁴

CSI 2 h

Locks the keyboard.

CSI 2 l

Unlocks the keyboard.

CSI 3 h

Enable display of control characters as symbols.

CSI 3 l

Disable display of control characters (execute control characters).

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 42 h** Changes emulation to Wyse 60.
- 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).
- CSI ? 5 h** Light background, dark text.
- CSI ? 5 l** Dark background, light text.
- 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. ²³⁴
- CSI ? 9 l** Status line on. ²³⁴
- CSI 1 \$ ~** Display local status line. ²³⁴
- CSI 2 \$ ~** Display host-writable status line. ²³⁴

CSI 0 \$ }

Data sent to screen's data area. ²³⁴

CSI 1 \$ }

Data sent to host-writable status line. ²³⁴

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.

CSI ? 25 l

Cursor off.

CSI ? 38 h or **GS** or **ESC 1**

Enters Tektronix 4014 graphics mode.

CSI ? 38 l or **CAN** or **ESC 2**

Exits Tektronix 4014 graphics mode.

CSI ? 66 h

Cursor keys return application codes. ²³⁴

CSI ? 66 l

Cursor keys return cursor codes. ²³⁴

CSI ? 67 h

Backspace keys sends **BS** control code. ³⁴

CSI ? 67 l

Backspace keys sends **DEL** control code. ³⁴

CSI ? 69 h

Enables vertical split screen mode. ⁴

CSI ? 69 l

Disables vertical split screen mode. ⁴

CSI ? 95 h

Do not clear screen when column mode changes. ³⁴

CSI ? 95 1

Clear screen when column mode changes. ³⁴

CSI ? 109 h

Set CapsLock keyboard state.

CSI ? 109 1

Clear CapsLock keyboard state.

ESC % ! 0

Enters Tektronix 4100 graphics mode.

CSI 0 SP q or **CSI 1 SP q** or **CSI 2 SP q**

Select block cursor. ²³⁴

CSI 3 SP q or **CSI 4 SP q**

Select underscore cursor. ²³⁴

CSI n \$ |

Set number of columns to **n**. ³⁴

CSI n t or **CSI n * |**

Set number of rows to **n**. ³⁴

CSI n , q

Sets the terminal ID returned in response to the DA1 command. Valid values for **n** are: 0 = VT100, 1 = VT101, 2 = VT102, 5 = VT220 and 6 = VT320. ³⁴

CSI n * x

Selects if visual attributes which are modified by the DECCARA or DECRARA commands are contained within the rectangular area defined by the beginning and ending positions. If **n** = 0 or 1 then beginning and ending positions indicate locations in the character stream displayed on the screen; if **n** = 2, then the beginning and ending positions indicate the upper-left and lower-right corners of a rectangle. ^L

CSI ? 9 h

Enable mouse reporting. ^L

CSI ? 9 1

Disable mouse reporting. ^L

CSI n = L

Set erase mode. If **n** = 0, new lines and clear screen are filled with the current background color. Otherwise, new lines and clear screen fill with the screen background color. ^S

ESC =

Numeric pad returns application codes.

ESC > Numeric pad returns numbers.

Character Set Selection

ESC (B

Sets character set G0 to ASCII (default). ^{1234L}

ESC) B

Sets character set G1 to ASCII (default). ^{1234L}

ESC * B

Sets character set G2 to ASCII. ^{234L}

ESC + B

Sets character set G3 to ASCII. ^{234L}

ESC (<

Sets character set G0 to UPSS (the character set selected in the terminal settings). ^{234L}

ESC) <

Sets character set G1 to UPSS (the character set selected in the terminal settings). ^{234L}

ESC * <

Sets character set G2 to UPSS (the character set selected in the terminal settings - this is the default). ²³⁴

ESC + <

Sets character set G3 to UPSS (the character set selected in the terminal settings). ²³⁴

ESC (A

Sets character set G0 to ISO-Latin1. ^{1L}

ESC) A

Sets character set G1 to ISO-Latin1. ^{1L}

ESC - A

Sets character set G1 to ISO-Latin1. ²³⁴

ESC . A

Sets character set G2 to ISO-Latin1. ²³⁴

ESC / A

Sets character set G3 to ISO-Latin1. ²³⁴

ESC (0

Sets character set G0 to graphics. 1234L

ESC) 0

Sets character set G1 to graphics. 1234L

ESC * 0

Sets character set G2 to graphics. 234

ESC + 0

Sets character set G3 to graphics. 234

ESC (s

Sets character set G0 to scientific. 234

ESC) s

Sets character set G1 to scientific. 234

ESC * s

Sets character set G2 to scientific. 234

ESC + s

Sets character set G3 to scientific. 234

ESC (% 5

Sets character set G0 to Multinational. 1234

ESC) % 5

Sets character set G1 to Multinational. 1234

ESC * % 5

Sets character set G2 to Multinational. 234

ESC + % 5

Sets character set G3 to Multinational. 234

ESC (κ or ESC (υ

Sets character set G0 to codepage 437. ^L

ESC) κ or ESC) υ

Sets character set G1 to codepage 437. ^L

ESC (VT

Sets character set G0 to codepage 437. ^{LS}

ESC) VT

Sets character set G1 to codepage 437. ^{LS}

ESC (FF

Sets character set G0 to upper half of codepage 437. ^{LS}

ESC) FF

Sets character set G1 to upper half of codepage 437. ^{LS}

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. ^{1234LS}

S0

Invoke character set G1 into GL. ^{1234LS}

ESC ~

Invokes character set G1 into GR (the GR set corresponds to character codes in the range of hex A0-FE). ²³⁴

ESC n

Invokes character set G2 into GL. ²³⁴

ESC }

Invokes character set G2 into GR (default). ²³⁴

ESC o

Invokes character set G3 into GL. ²³⁴

ESC |

Invokes character set G3 into GR. ²³⁴

SS2 or ESC N

Invokes character set G2 into GL for the next character only. ²³⁴

SS3 or ESC o

Invokes character set G3 into GL for the next character only. ²³⁴

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.

VTSame 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.

CSI n TMoves cursor down *n* lines in the same column. If the cursor is moved to the last line of the scrolling region, the screen will scroll up. ^S**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.

CSI n SMoves the cursor up *n* lines in the same column. If the cursor is moved to the first line of the scrolling region, the screen will scroll down. ^S**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. 1234LS**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. 1234LS**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* dMoves cursor to line *n*. 1234LS**CSI *n* G** or **CSI *n* ^**Moves cursor to column *n*. 1234LS**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. 1234LS

CSI 3 g or **CSI 5 w**

Clears all tab stops. 1234LS

CSI ? 5 w

Sets tab stops every 8th column. 234

CSI *n* IMove forward *n* tab stops. 234**CSI *n* Z**Move backward *n* tab stops. 234**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. 234**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. 234**CSI *n* SPACE P**Display page *n*. 234**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. 1234LS**ESC 7** or **CSI s** ^{ALS}

Save state (cursor position, video attribute, character set, autowrap, origin mode and protect mode).

ESC 8 or **CSI u** ^{ALS}

Restore state.

CSI left ; right s

Sets the left and right margins to define a *horizontal* scrolling region. This command only works when vertical split screen mode is enabled.

ESC 6

Move cursor left one column. If the cursor is at the left margin, all data within the margin scrolls right one column. The column that shifted past the right margin is lost.⁴

ESC 9

Move cursor right one column. If the cursor is at the right margin, all data within the margin scrolls left one column. The column that shifted past the left margin is lost.⁴

ESC 8

Fill screen with upper-case "E".

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).

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. ^{234LS}

CSI n @

Insert *n* blank characters beginning at the current cursor position. ^{234LS}

CSI n L

Insert *n* blank lines beginning at the cursor line. ^{1234LS}

CSI n P

Delete *n* characters beginning at the current cursor position. ^{1234LS}

CSI n M

Delete *n* lines beginning at the cursor line. ^{1234LS}

CSI n ' }

Insert *n* columns into the scrolling region beginning at the column that has the cursor. ⁴

CSI n ' ~

Delete *n* columns from the scrolling region beginning at the column that has the cursor. ⁴

CSI top ; left ; bottom ; right ; attr1 ; ... ; attrn \$ x

Changes visual attributes in an area defined by *top*, *left*, *bottom*, *right*. The area to be changed is either a character stream or rectangle as defined by the DECSACE command.

See [ANSI Attribute Code Table](#) for the attribute codes. ⁴

CSI top ; left ; bottom ; right ; attr1 ; ... ; attrn \$ t

Reverse visual attributes in an area defined by *top*, *left*, *bottom*, *right*. The area to be changed is either a character stream or rectangle as defined by the DECSACE command.

See [ANSI Attribute Code Table](#) for the attribute codes. ⁴

CSI top ; left ; bottom ; right \$ z

Erase characters in rectangle defined by *top*, *left*, *bottom*, *right*. ⁴

CSI top ; left ; bottom ; right \$ {

Erase unprotected characters in rectangle defined by *top*, *left*, *bottom*, *right*. ⁴

CSI ch ; top ; left ; bottom ; right \$ x

Fill rectangle defined by *top*, *left*, *bottom*, *right* with character *ch* (ASCII code). ⁴

CSI stop ; sleft ; sbottom ; sright ; spg ; dtop ; dleft ; dpd \$ v

Copy rectangle defined by **stop**, **left**, **sbottom**, **sright**, from page **spg** to page **dpg** at location **dtop**, **dleft**.⁴

Video Attributes

CSI n ; n . . . m

Selects video attributes and/or character foreground and background colors according to the [ANSI Attribute Code Table](#). 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.²³⁴

CSI 1 " q

Protected mode. Characters received after this command are not erasable using the “erase unprotected characters” command.²³⁴

ESC # 3

Double-high line top. Causes the line containing the cursor to display the top half of a double-high line.¹²³⁴

ESC # 4

Double-high line bottom. Causes the line containing the cursor to display the bottom half of a double-high line.¹²³⁴

ESC # 5

Single-width line. Causes the line containing the cursor to display normal width characters.¹²³⁴

ESC # 6

Double-width line. Causes the line containing the cursor to display double width characters.¹²³⁴

CSI = n F

Sets the current normal foreground color to **n**. Refer to “AccuTerm Programming” chapter for color values.^S

CSI = n G

Sets the current normal background color to **n**.^S

CSI = n H

Sets the current reverse foreground color to **n**.^S

CSI = n I

Sets the current reverse background color to **n**.^S

ESC] P nrrggb

Set Linux Console palette. **N** is a single hex digit indicating which palette entry to set; **rrggb** are the hex RGB color values to be set. ^L

ESC] R

Reset Linux Console palette. ^L

Printer Control**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 10 i

Prints the screen display ignoring the print extent in effect. ³⁴

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. When the “Auto-print works just like a VT terminal” setting is enabled, a line is printed from the screen when the cursor moves off that line with an *LF*, *FF*, or *VT* control code, or an autowrap occurs. Otherwise auto-print mode works just like Wyse auto-print mode.

CSI 4 i or CSI ? 4 i

Cancel transparent or auto print mode.

CSI 2 i

Send screen to host. ²³⁴

Terminal Reports**CSI 0 c or ESC z**

Request primary device attributes. Depending on the current emulation and terminal ID in effect, AccuTerm will respond: ^{1234L}

VT100: **CSI ? 1 ; 2 c**

VT220: **CSI ? 6 2 ; 1 ; 2 ; 6 ; 8 c**

VT320: **CSI ? 6 3 ; 1 ; 2 ; 6 ; 8 c**

VT420: **CSI ? 6 4 ; 1 ; 2 ; 6 ; 8 c**

Linux: **CSI ? 6 c**

CSI > 0 c

Request secondary device attributes. AccuTerm will respond: ²³⁴

CSI > 41 ; 4 ; 1 c

CSI = 0 c

Request tertiary device attributes. AccuTerm will respond (~~xxxxxxxx~~ is the product serial number in hexadecimal): ²³⁴

DCS ! |1 xxxxxxxx ST

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 ? 6 n

Request cursor position and page. AccuTerm will respond:

CSI line ; column ; page R

CSI ? 1 5 n

Request printer status. If a printer is defined for the current configuration, AccuTerm will respond: ²³⁴

CSI ? 1 0 n

If no printer is defined the response will be:

CSI ? 1 3 n

CSI ? 2 5 n

Request status of user-defined keys. AccuTerm will respond: ²³⁴

CSI ? 2 0 n

CSI ? 2 6 n

Request keyboard status. AccuTerm will respond: ²³⁴

CSI ? 2 7 ; 0 ; 0 ; 0 n

CSI ? 5 5 n

Request locator status. If the mouse is currently enabled, AccuTerm will respond: ²³⁴

CSI ? 5 0 n

If the mouse is not enabled, the response will be:

CSI ? 5 3 n

CSI ? 5 6 n

Request locator device type. If the mouse is currently enabled, AccuTerm will respond: ²³⁴

CSI ? 5 7 ; 1 n

If the mouse is not enabled, the response will be:

CSI ? 5 7 ; 0 n

CSI ? 6 2 n

Request macro space. AccuTerm will respond: ²³⁴

CSI 0 * {

CSI ? 6 3 ; id n

Request memory checksum. AccuTerm will respond: ²³⁴
DCS id ! ~ 0 0 0 0 ST

CSI ? 7 5 n

Request data integrity. AccuTerm will respond: ²³⁴
CSI ? 7 0 n

CSI " v

Request displayed extent. AccuTerm will respond: ³⁴
CSI rows ; columns ; 1 ; 1 ; page " w

CSI n \$ p

Request state of ANSI mode *n*. AccuTerm will respond: ³⁴
CSI n ; value \$ y

CSI ? n \$ p

Request state of private mode *n*. AccuTerm will respond: ³⁴
CSI ? n ; value \$ y

CSI + x

Request function key free memory. AccuTerm will respond: ³⁴
CSI 804 ; 804 + y

CSI 1 \$ w

Request cursor information report. AccuTerm will respond with a string encoded with the current cursor state including position, page, attribute and character set. The string returned can be use to restore the cursor state. The response string is: ³⁴
DCS 1 \$ u string ST

DCS 1 \$ t string ST

Restore cursor state. Use **string** returned from previous command. ³⁴

CSI 2 \$ w

Request tabstop report. AccuTerm will respond with a string encoded with the current tab settings. The string returned can be use to restore the tabs. The response string is: ³⁴
DCS 2 \$ u string ST

DCS 2 \$ t string ST

Restore tabstops. Use **string** returned from previous command. ³⁴

DCS \$ q setting ST

Request setting. **Setting** is formed by taking the last one or two non-numeric characters of an ANSI command. The response is: ³⁴

DCS 0 \$ r string ST

You can use the response **string** to restore the setting; Simply add *CSI* to the beginning of the string and send it back to the terminal.

CSI 1 \$ u

Request terminal state. The complete terminal state (cursor position, character set,

attributes, screen size, key lock, etc.) is encoded into a string and returned to the host.

You can use this string to restore the state later. The response is: ³⁴

DCS 1 \$ s *string* ST

DCS 1 \$ p *string* ST

Restore terminal state. ***String*** is the value returned from the previous command. ³⁴

Programming Function Keys

The AccuTerm VT220, VT320, VT420, Linux Console and SCO Console emulations support 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](#), and the sequence for that key is specified by ***value***. ***Value*** is specified in hexadecimal.

VT320 and VT420 emulations provide an extended method to program function and editing keys:

DCS " *x* *key* / *mod* / 100 / *value* / 0 ; . . . ST

The key to be programmed is specified by ***key*** according to the [ANSI Extended Key Table](#), and any modifier keys (SHIFT, CTRL, etc.) are specified by ***mod***, according to the same table. The sequence for the key is specified by ***value*** in hexadecimal.

In addition, the VT420 emulation has user-definable macros. Use the following command to define macros.

DCS *id* ; *dt* ; *fn* ; *en* ! *z* *value* ST

Id is a macro identifier, which must be in the range of 0 to 63. If *dt* is 1, then all current macros are deleted before the new macro is defined. *en* specifies the encoding: 0 for ASCII text, 1 for hexadecimal. *Value* is the macro contents in the specified encoding format.

Invoking a macro has the same effect as if the terminal had received the macro contents from the host. The following command may be use to invoke a defined macro:

CSI *id* * *z* ⁴

The SCO Console emulation uses the following sequence to program function keys:

ESC Q *key* *delim* *value* *delim*

key is a single ASCII character which designates which key to program. Function keys F1 to F12 are selected by specifying 0 to ; . For SHIFT+F1 to SHIFT+F12 use < to g. For CTRL+F1 to CTRL+F12 use h to s. For CTRL+SHIFT+F1 to CTRL+SHIFT+F12 use t to _. *Delim* is a single character delimiter which encloses *value*.

CSI 2 + *z*

Restores default values to programmed keys. ³⁴

To program the Answerback message, use the following device control sequence:

DCS 1 v value ST

Note: the "Map F11 through F20 ..." check box in the keyboard settings 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 F	Enter line graphics mode.
ESC G	Exit line graphics mode.
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 *line col*
Cursor position.

3.4.1 ANSI Tables

3.4.1.1 ANSI Attribute Code Table

The **ANSI Attribute Code Table** defines the attribute code character (*n*) used by the ANSI emulations to assign visual attributes.

Code (<i>n</i>)	Attribute type
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

Code (<i>n</i>)	Attribute type
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

3.4.1.2 ANSI Function Key Table

The **ANSI Function Key Table** defines the key code (**key**) used by the ANSI emulations to program function keys.

Key to program	key
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

3.4.1.3 ANSI Extended Key Table

The **ANSI Extended Key Table** defines the key code (**key**) and modifier keys (**mod**) used by the ANSI emulations to program extended function and keypad keys.

Key to program	key	Modifier keys	mod
F1	112	none	0
F2	113	SHIFT	2
F3	114	ALT	3
F4	115	ALT+SHIFT	4
F5	116	CTRL	5
F6	117	SHIFT+CTRL	6

Key to program	<i>key</i>	Modifier keys	<i>mod</i>
F7	118	ALT+CTRL	7
F8	119	ALT+CTRL+SHIFT	8
F9	120		
F10	121		
F11	122		
F12	123		
BKSP	14		
TAB	16		
INS	75		
DEL	76		
HOME	80		
END	81		
PGUP	85		
PGDN	86		
LEFT	79		
RIGHT	89		
UP	83		
DOWN	84		
ESC	110		
ENTER	42		
KPD ENTER	108		

3.4.1.4 ANSI Key Code Table

The **ANSI Key Code Table** describes the default character sequence transmitted by the function, keypad and other special keys on the PC keyboard and the corresponding real terminal key.

PC Key	Terminal Key	Sequence
BKSP	BKSP	<i>BS</i>
SHIFT+BKSP	SHIFT+BKSP	<i>BS</i>
TAB	TAB	<i>HT</i>
SHIFT+TAB	SHIFT+TAB	<i>CSI z</i>
INS		<i>CSI 2 ~</i>
SHIFT+INS		<i>CSI 2 ~</i>
DEL	DEL	<i>DEL</i>
SHIFT+DEL	DEL LINE	<i>DEL</i>
HOME	HOME	<i>CSI H</i>
SHIFT+HOME	SHIFT+HOME	<i>CSI H</i>
END		<i>CSI 1 ~</i>
SHIFT+END		<i>CSI 1 ~</i>
PGUP	PGUP	<i>CSI 5 ~</i>
SHIFT+PGUP	SHIFT+PGUP	<i>CSI 5 ~</i>
PGDN	PGDN	<i>CSI 6 ~</i>
SHIFT+PGDN	SHIFT+PGDN	<i>CSI 6 ~</i>
LEFT	LEFT	<i>CSI D</i>
SHIFT+LEFT	SHIFT+LEFT	<i>CSI D</i>

PC Key	Terminal Key	Sequence
RIGHT	RIGHT	<i>CSI</i> C
SHIFT+RIGHT	SHIFT+RIGHT	<i>CSI</i> C
UP	UP	<i>CSI</i> A
SHIFT+UP	SHIFT+UP	<i>CSI</i> A
DOWN	DOWN	<i>CSI</i> B
SHIFT+DOWN	SHIFT+DOWN	<i>CSI</i> B
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>
F1	PF1	<i>SS3</i> P
F2	PF2	<i>SS3</i> Q
F3	PF3	<i>SS3</i> R
F4	PF4	<i>SS3</i> S
F5		<i>CSI</i> M
F6	F6	<i>CSI</i> 1 7 ~
F7	F7	<i>CSI</i> 1 8 ~
F8	F8	<i>CSI</i> 1 9 ~
F9	F9	<i>CSI</i> 2 0 ~
F10	F10	<i>CSI</i> 2 1 ~
F11	F11	<i>CSI</i> 2 3 ~
F12	F12	<i>CSI</i> 2 4 ~
CTRL+F1	F11	<i>CSI</i> 2 3 ~
CTRL+F2	F12	<i>CSI</i> 2 4 ~
CTRL+F3	F13	<i>CSI</i> 2 5 ~
CTRL+F4	F14	<i>CSI</i> 2 6 ~
CTRL+F5	F15	<i>CSI</i> 2 8 ~
CTRL+F6	F16	<i>CSI</i> 2 9 ~
CTRL+F7	F17	<i>CSI</i> 3 1 ~
CTRL+F8	F18	<i>CSI</i> 3 2 ~
CTRL+F9	F19	<i>CSI</i> 3 3 ~
CTRL+F10	F20	<i>CSI</i> 3 4 ~

Note: when using 7 bit controls, *CSI* is sent as *ESC* [and *SS3* is sent as *ESC* o.

3.4.2 ANSI Programming Notes

Functions tagged with the following super-script symbols are only valid in the designated emulation:

- 1 - VT100
- 2 - VT220
- 3 - VT320
- 4 - VT420
- A - ANSI BBS
- L - Linux Console
- S - SCO Console

3.5 Pick PC Console Programming

The Pick Operating System on the IBM PC has its own terminal type for the system console (term type `x`). AccuTerm emulates the Pick PC console thus allowing you to use the same terminal codes for all your terminals in a Pick PC environment. The advantage of using this emulation is that it gives you direct access to legacy VGA colors and attributes.

Operating Modes

ESC * Y

Set the video mode to 80 by 25 monochrome mode. This command is ignored by AccuTerm.

ESC * J

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

In Pick PC Console emulation there are several different ways to set the video attributes. The first is through escape sequences that start and stop individual attributes like "Underline", "Reverse", etc. The second is by sending the desired foreground and background colors. Note that the PC Console treats the "Dim" attribute as the protected attribute. To change video attributes, 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 the [PC Console Color Code Table](#).

Protected Attributes

The Pick PC Console emulation 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 contain protected characters, the auto scroll mode is turned off, and the clear screen functions clear only the un-protected characters.

*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.

3.5.1 PC Console Color Code Table

The **PC Console Color Code Table** defines the color code character (*n*) used by the Pick PC Console emulation to assign the foreground and background colors.

Color	Background	Bright foreground	Dim foreground
White	!)	9
Yellow	"	*	:
Magenta	#	+	;
Red	\$,	<
Cyan	%	-	=
Green	&	.	>
Blue	'	/	?
Black	(0	@

3.6 ASCII Control Codes

ASCII Control Codes

The **ASCII Control Codes** chart lists the ASCII 7-bit control codes, along with their decimal and hexadecimal values and the keyboard keys used to produce the code. The ANSI 8-bit control codes are shown at the end of the table along with their 7-bit equivalents.

ASCII 7-bit control char	Decimal	Hex	Key
<i>NUL</i>	0	00	
<i>SOH</i>	1	01	CTRL+A
<i>STX</i>	2	02	CTRL+B
<i>ETX</i>	3	03	CTRL+C
<i>EOT</i>	4	04	CTRL+D
<i>ENQ</i>	5	05	CTRL+E
<i>ACK</i>	6	06	CTRL+F
<i>BEL</i>	7	07	CTRL+G
<i>BS</i>	8	08	CTRL+H
<i>HT</i>	9	09	CTRL+I
<i>LF</i>	10	0A	CTRL+J
<i>VT</i>	11	0B	CTRL+K
<i>FF</i>	12	0C	CTRL+L
<i>CR</i>	13	0D	CTRL+M
<i>SO</i>	14	0E	CTRL+N
<i>SI</i>	15	0F	CTRL+O
<i>DLE</i>	16	10	CTRL+P
<i>DC1</i>	17	11	CTRL+Q
<i>DC2</i>	18	12	CTRL+R
<i>DC3</i>	19	13	CTRL+S
<i>DC4</i>	20	14	CTRL+T
<i>NAK</i>	21	15	CTRL+U
<i>SYN</i>	22	16	CTRL+V
<i>ETB</i>	23	17	CTRL+W
<i>CAN</i>	24	18	CTRL+X
<i>EM</i>	25	19	CTRL+Y
<i>SUB</i>	26	1A	CTRL+Z
<i>ESC</i>	27	1B	CTRL+[
<i>FS</i>	28	1C	CTRL+\
<i>GS</i>	29	1D	CTRL+]
<i>RS</i>	30	1E	CTRL+^
<i>US</i>	31	1F	CTRL+_
<i>DEL</i>	127	7F	
ANSI 8-bit control char	Decimal	Hex	7-bit equivalent
<i>IND</i>	132	84	
<i>NEL</i>	133	85	

ASCII 7-bit control char	Decimal	Hex	Key
<i>SSA</i>	134	86	
<i>ESA</i>	135	87	
<i>HTS</i>	136	88	
<i>HTJ</i>	137	89	
<i>VT8</i>	138	8A	
<i>PLD</i>	139	8B	
<i>PLU</i>	140	8C	
<i>RI</i>	141	8D	
<i>SS2</i>	142	8E	
<i>SS3</i>	143	8F	<i>ESC</i> o
<i>DCS</i>	144	90	<i>ESC</i> P
<i>PU1</i>	145	91	
<i>PU2</i>	146	92	
<i>STS</i>	147	93	
<i>CCH</i>	148	94	
<i>MW</i>	149	95	
<i>SPA</i>	150	96	
<i>EPA</i>	151	97	
	152	98	
	153	99	
	154	9A	
<i>CSI</i>	155	9B	<i>ESC</i> [
<i>ST</i>	156	9C	<i>ESC</i> \
<i>OSC</i>	157	9D	
<i>PM</i>	158	9E	
<i>APC</i>	159	9F	

3.7 Customizing the Installation Process

The AccuTerm 8 installation process can be customized in several ways: default values can be supplied for any of the installation wizard prompts, the installer can run in "silent mode", you can supply your own files to be installed along with the AccuTerm 8 program files, and the MSI can be extracted to facilitate deployment using Active Directory and Group Policy.

The AccuTerm 8 installer is distributed as a single EXE file. The installer can be placed in any folder that is accessible to the user's PC: local hard drive, portable drive such as a USB drive, network share, CD-ROM, etc. The EXE wraps an MSI install file and launches the MSI installer (msiexec) with appropriate options based on the install type selected.

Supplying default values for installation prompts

To supply default values for installation prompts, you need to create a text file called 'setup.ini' and place it in the same directory that the AccuTerm 8 installation EXE file resides. A sample 'setup.ini' file is shown below:

```
; Custom setup for AccuTerm 8
;
; Use atwin_8.0.1000.exe /silent or /verysilent for a "quiet" unattended
installation
;
; Items which can be defined:

[Defaults]
;MAINDIR=Program Files\Atwin80
;HOMEDIR=My Documents
;DESKTOPSHORTCUT=Y
;ALLUSERS=Y
;AUTOACTIVATE=Y
;AUTOUPGRADE=Y
;CUSTOMFEATURECODE=xxxx
;LICENSEKEY=xxxx-xxxx-xxxx-xxxx-xxxx-xxxx
;KEYTYPE=1
;EMAIL=you@company.com

[CustomFiles]
;Main=folder containing extra files copied to the main program directory
;Images=folder containing extra image files copied to the Images sub
directory
;Home=folder containing files to copy to the default "home" directory

[CustomShortcut]
;Name=name of shortcut
;Target=path to .exe, .atcf, .atly or .atsc file (can use %HOMEDIR% or %
MAINDIR% prefix for path)
;Icon=path to file containing icon (can use %HOMEDIR% or %MAINDIR%
prefix for path)
;Index=n (index of icon, if icon is contained in an exe file with
multiple icons)
;State=n (initial window state: 1=normal, 2=minimized, 3=maximized)
```

To use the 'setup.ini' sample shown above, remove the leading ';' from any items you need to define, and enter the value you want to use after the equal sign. Remove the comment text from any lines that you

define. Any items that are not defined will use the default for that setting.

The `MAINDIR` entry may begin with the text "Program Files" which will be replaced during the installation with the appropriate directory for program files (usually `C:\Program Files` or `C:\Program Files (x86)`).

The `HOMEDIR` entry may begin with the text "My Documents" which will be replaced during installation with the public Documents directory (Normal install) or the current user's Documents directory (Personal install). Subdirectories are allowed.

Set `DESKTOPSHORTCUT` to `Y` to create shortcut to AccuTerm 8 on the desktop or `N` to omit the desktop shortcut.

Set `ALLUSERS` to `Y` to install for all users, or `N` to install only for the current user.

Set `AUTOACTIVATE` to `Y` if you want to activate the AccuTerm license automatically after installation. You must also specify the `LICENSEKEY` and `KEYTYPE` settings to use automatic activation. For device-based licensing, set `KEYTYPE=1` and for user-based licensing set `KEYTYPE=0`, and also specify the `EMAIL` associated with the license key.

The `AUTOUPGRADE` item is used to skip the "AccuTerm 8 Upgrade" dialog that is shown on first use of AccuTerm 8 when an earlier version of AccuTerm is detected. Set this item to "N" to suppress the dialog and ignore all previous settings and files. Set to "Y" to migrate all possible preferences and files from the earlier AccuTerm version. Set this item to a numeric value (a bit mask) to specify which category of settings or files are to be automatically migrated from the earlier version of AccuTerm. The values for each desired category can be added together to form the bit mask.

- 1 - migrate user's personal custom menu
- 2 - migrate application custom menu
- 4 - migrate settings from the current `atwin.ini` file
- 8 - migrate preferences (stored in the `HKEY_CURRENT_USER` section of the registry)
- 16 - migrate SSH keys
- 32 - migrate custom settings from the `atwin.ini` file
- 64 - migrate session files from program directory to home directory
- 256 - migrate `cfg`, `lyt`, `atcf`, `atly`, `script` and image files found in the old program directory

In the `[CustomShortcut]` section, the `Target` and `Icon` entries may begin with the tokens `%MAINDIR%` and `%HOMEDIR%`. These tokens will be replaced with the appropriate directory during installation.

Installing user-supplied files

The AccuTerm 8 Installer can install user supplied files after installing the AccuTerm program files. To install user supplied files, create a sub-directory below the directory where the installation EXE resides called 'VARFILES'.

To copy user supplied files to the AccuTerm program directory, create a sub-directory under `VARFILES` called 'MAIN'. Place any files that need to be copied to the AccuTerm program directory in this directory. Add the entry "`Main=VARFILES\MAIN`" in the `[CustomFiles]` section of `setup.ini`.

To copy user supplied image files to the AccuTerm program Images sub-directory, create a sub-directory under `VARFILES` called 'IMAGES' and place the image files in this sub-directory. Add the entry "`Images=VARFILES\IMAGES`" in the `[CustomFiles]` section of `setup.ini`.

To copy user supplied files to the user's "home" directory (the directory where the session files are

saved), create a sub-directory under VARFILES called 'HOME'. Place files to copy to the "home" directory in this directory. Add the entry "Home=VARFILES\HOME" in the [CustomFiles] section of setup.ini.

Note: in prior versions of AccuTerm, the VARFILES directory could contain a directory tree to copy to the program or home directories. Directory trees are no longer supported. Only individual files can be copied. Also, prior versions of AccuTerm allowed a user-specified secondary installation program to be executed during the install process. This is no longer supported.

Installing a pre-configured version of AccuTerm

It is possible to create a custom installation package that installs a pre-configured version of AccuTerm. AccuTerm's initial state is defined by settings stored in the 'atwin80.ini' file in the AccuTerm program directory. Custom master menu/toolbar is defined by the 'menu.json' file in the AccuTerm program directory. Session settings are stored in session configuration files (.atsp files), stored in the user's (or public) "home" directory. By taking advantage of the ability of the AccuTerm installer to install user supplied files (see above) you can create a pre-configured installation package.

Note that the atwin80.ini file only specifies initial preferences and settings. The settings in atwin80.ini are used to initialize the working settings which are saved in the registry the first time the program is run.

Here is a sample atwin80.ini file to use as a basis for your custom settings:

```
[Atwin]
Version=8.0

[Options]
ToolBar=True
StatusLine=True
ShortcutNewWindow=False
AutoClose=False
NoCloseWarning=False
NoConnectWizard=False
NoWelcomePage=False
TrackKeyboardState=False
FuncBar=False
FuncBarStyle=0
FuncBarPos=0
TitleFormat=4
RecentListSize=4

[Path]
Capture=" "

[Xfer]
DefaultDir=" "

[Custom]
Icon=" "
Title=" "
Splash=" "
```

Most of the 'atwin80.ini' settings are obvious. For those values which are not True/False, please refer to the property of the same name in the AccuTerm object. The [Custom] Icon and Splash settings are file names. These files are usually stored in the AccuTerm program directory - in that case you do not need to include a full path. You can disable the splash screen by setting Splash=No.

Unattended Installation

The AccuTerm 8 installer uses command-line options to perform an unattended installation.

The "/SILENT" command-line option bypasses all of the installation prompts, but does display a progress bar during installation:

```
atwin_8.0.1000.exe /SILENT
```

The "/VERYSILENT" command-line option bypasses all of the prompts and does not display the progress bar:

```
atwin_8.0.1000.exe /VERYSILENT
```

Extracting the MSI file from the EXE installation file

To extract the MSI file from the EXE installer, run the EXE with the /EXTRACTMSI command line option:

```
atwin_8.0.1000.exe /EXTRACTMSI
```

The EXE installer will prompt for a target directory where the MSI file will be stored.

Installing AccuTerm directly using MSI

Once the MSI file has been extracted from the EXE, you can use the `msiexec` command to install directly from the MSI file. You may specify certain MSI properties on the `msiexec` command line. Common properties are `INSTALLDIR`, `ALLUSERS`, and `MSIINSTALLPERUSER`. In order to use `setup.ini` as described above, specify the `SETUPINI` property on the command line (the value is the name of the `setup.ini` file). For example:

```
msiexec /i atwin80.msi ALLUSERS="1" SETUPINI="setup.ini"
```

MSI installation notes

- for a silent installation, include the `/qb` or `/qn` options on the `msiexec` command line. The `/qb` option will display the installation progress window; the `/qn` option will not display anything.
- the `MAINDIR` item in `setup.ini` is ignored when installing from MSI. Instead, specify `INSTALLDIR="<path>"` on the command line. When performing a silent install, the path must be absolute.

4 Features for MultiValue Users

4.1 MultiValue File Transfer

To make the most effective use of AccuTerm with your MultiValue based host system, you should install the AccuTerm host programs. These programs implement the Kermit and ASCII file transfer protocols, coordinate the transfer of data between your computer and the host, import and export data in a variety of formats and provide a simple host-to-host transfer facility. See [Installing MultiValue Host Programs](#) for information on installing the host programs.

In order to understand how to best use AccuTerm's MultiValue file transfer features, the differences between the PC and MultiValue file systems must be examined. Traditional PC files may be programs, documents or data files; in the typical MultiValue system, programs and documents are stored as individual items (records) in a file, whereas a data file is collection of related items (records) in a file, with an associated file dictionary.

When exchanging data between your computer and your MultiValue host, it is important to realize the difference between "document files" and "data files".

MultiValue File Transfer Wizard

Use the File Transfer Wizard to simplify the process of uploading or downloading data and documents between your PC and MultiValue host. The wizard uses a series of simple questions to guide you through the file transfer process. You can use the wizard to help you upload or download data or document files.

MultiValue File Transfer Utilities

When transferring "document files" between the PC and MultiValue host, use AccuTerm's **FT** utility. This utility transfers MultiValue *items* to PC *files* and vice versa. The key concept is that each MultiValue *item* becomes a separate *file* on the PC.

When transferring "data files" between your PC and your MultiValue host, use the **FTD** utility. This utility transfers a set of MultiValue *items* to a single PC *file* and vice versa. The key concept is that a set of *items* on the host becomes a single *file* on the PC.

You can also use AccuTerm to transfer data between two MultiValue hosts. The **FTPICK** utility is used to transfer data from one MultiValue machine to another, either directly (using two concurrent AccuTerm sessions), or using an intermediate PC file.

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 file.

MultiValue File Transfer Functions:

- [Installing MultiValue Host Programs](#)
- [Updating the MultiValue Host Programs](#)
- [Downloading documents to your PC](#)
- [Uploading documents from your PC](#)
- [Downloading data files to your PC \(export\)](#)
- [Uploading data files from your PC \(import\)](#)
- [Transferring data between two MultiValue hosts](#)
- [Integrating file transfer functions into MultiValue applications](#)

4.1.1 Installing the MultiValue Host Programs

To install the MultiValue host programs, perform the following steps.

1. Open a terminal session and connect to your MultiValue host system. If you have an existing **ACCUTERM** account, log in to the **ACCUTERM** account (a different account may be used, but it is highly recommended that you use a dedicated **ACCUTERM** account). Otherwise log in to your system administration account (**SYS**PROG, **DM**, **QMSYS**, **UV**, **UD**, etc.) and create an **ACCUTERM** account (you may need to ask your system administrator to create the **ACCUTERM** account for you).
2. Check the following items before beginning the installation:
 - a) Make sure you are at a command prompt (TCL, ECL, jsh, etc.)
 - b) If your system is running on a Unix, Linux or AIX host, make sure that the **CTRL+A** control character can be input. Check by typing `stty -a` at a *nix prompt – look for **^A** - if you find it, either disable that function or map it to a different control character. For example if you see something like `susp = ^A` you can type `stty susp \^-` to disable the suspend function.
 - c) For jBase systems, make sure you have a valid terminal type assigned.
3. While at the command prompt use **MultiValue ? Host Programs ? Install** from the main menu to start the host program installation wizard. Confirm that you want to install the host programs by clicking the **Yes** button.
4. The host program installation wizard will perform some checks to identify the type of MultiValue platform you are running, verify that you are at a TCL prompt and turn off case inversion. You will see meaningless text on the terminal screen during this process.
5. The installation wizard will display a list of supported MultiValue platforms. Select your platform from the list. If the wizard was able to identify the platform, it will be highlighted, but you can change it by clicking a different platform if necessary.
6. If your MultiValue platform is running on IBM's AIX operating system, check the **Host operating system is IBM AIX** check-box.
7. Click the **Install Host Programs** button to install a bootstrap PROC and BASIC program. You will see these scroll up the terminal screen as they are installed. Once the bootstrap programs are installed, the wizard will run them to install the core FTBP programs automatically.
8. After the core FTBP programs are installed, a dialog box will be displayed where you can select any optional program libraries (GUI, ObjectBridge, SUI) and sample programs and data that you want to install. If you are going to use **phiReport** (a premium AccuTerm feature), be sure to check the **All items needed for phiReport** check-box, or select the **Install all items** option.
9. If you have previously installed the ViaDuct host programs, the ViaDuct FT PROC conflicts with AccuTerm's FT program. If this conflict is detected, the AccuTerm installer will prompt if you want to rename the ViaDuct FT command as "FT.VIA". Answer **y** to rename.

An alternate method of installing the AccuTerm host programs is available in the event that the installation wizard is not compatible with your host. See [Alternative Host Program Installation](#).

See also:

[Updating the MultiValue Host Programs](#)

[Activating other accounts to use the AccuTerm host programs](#)

[Installing phiReport reporting tool](#)

4.1.1.1 Alternative Host Program Installation

To install the AccuTerm MultiValue host programs manually, perform the following steps.

1. Open a terminal session and connect to your MultiValue host system. If you have an existing **ACCUTERM** account, log in to the **ACCUTERM** account (a different account may be used, but it is highly recommended that you use a dedicated **ACCUTERM** account). Otherwise log in to your system administration account (**SYSPROG**, **DM**, **QMSYS**, **UV**, **UD**, etc.) and create an **ACCUTERM** account (you may need to ask your system administrator to create the **ACCUTERM** account for you).
2. Check the following items before beginning the installation:
 - a) Make sure you are at a command prompt (TCL, ECL, jsh, etc.)
 - b) If your system is running on a Unix, Linux or AIX host, make sure that the **CTRL+A** control character can be input. Check by typing `stty -a` at a *nix prompt – look for **^A** - if you find it, either disable that function or map it to a different control character. For example if you see something like `susp = ^A` you can type `stty susp \^-` to disable the suspend function.
 - c) For jBase systems, make sure you have a valid terminal type assigned.
 - d) Check the **Paste Options - end-of-line** setting ([Clipboard Settings](#)) and ensure that the **CR** option is selected. Ensure that the **Do not send line-end...** option is not selected.
3. While at the command prompt (TCL, ECL, jsh, etc.), click on the **Edit** menu and select **Paste From**.
4. Browse to the PICKBP folder.
5. Click on the install script for your platform (please see the readme.txt file in the PICKBP folder for the latest platform list):

FTPINST.TXT	generic MultiValue system
FTPINSADDS.TXT	Mentor, Mentor Pro, MOE
FTPINSAP.TXT	Advanced Pick (except PRO and AIX)
FTPINSAP_PRO.TXT	Advanced Pick / Pro
FTPINSCACHE.TXT	Intersystems Cache (except AIX)
FTPINS D3.TXT	D3 (NT, Linux, Unix except AIX)
FTPINSENH.TXT	Enhanced R83 (Altos, Fujitsu, R91, etc.)
FTPINSJB.TXT	jBASE 3.x (except AIX)
FTPINSJB4.TXT	jBASE 4.x (except AIX)
FTPINSJB5.TXT	jBASE 5.x (except AIX)
FTPINSMD.TXT	Reality
FTPINSMV.TXT	mvBase
FTPINSNR.TXT	Northgate Reality 9+
FTPINSOI.TXT	Revelation OpenInsight CTO
FTPINSON.TXT	ONWare MVON (except AIX)
FTPINSPWR95.TXT	Power 95
FTPINSQM.TXT	QM MV Database
FTPINSQM3.TXT	QM 3.x MV Database
FTPINSR83.TXT	Pick R83
FTPINSSEQ.TXT	Sequoia, mvEnterprise (except Ultimate emulation and AIX)
FTPINSUD.TXT	UniData (except AIX)
FTPINSULT.TXT	Ultimate Plus (all except SCO Unix and AIX)
FTPINSULT_SCO.TXT	Ultimate Plus on SCO Unix
FTPINSULTX.TXT	Early Ultimate (as implemented on mvEnterprise)
FTPINSUV.TXT	UniVerse (except AIX)
FTPINSVIS.TXT	UniVision

For hosts running on IBM AIX, use the following scripts:

FTPINSAP_AIX.TXT	Advanced Pick on AIX
FTPINSCACHE_AIX.TXT	Intersystems Cache on AIX
FTPINSDB3_AIX.TXT	D3 on AIX
FTPINSJB_AIX.TXT	jBASE 3.x on AIX
FTPINSJB4_AIX.TXT	jBASE 4.x on AIX
FTPINSJB5_AIX.TXT	jBASE 5.x on AIX
FTPINSON_AIX.TXT	ONWare MVON on AIX
FTPINSSEQ_AIX.TXT	Sequoia, mvEnterprise on AIX
FTPINSUD_AIX.TXT	UniData on AIX
FTPINSULT_AIX.TXT	Ultimate Plus on AIX
FTPINSULTX_AIX.TXT	Early Ultimate on AIX (as implemented on mvEnterprise)
FTPINSUV_AIX.TXT	UniVerse on AIX

- Click the Open button. AccuTerm will install a small bootstrap PROC and program which will be used to install the core file transfer programs. You will see these programs scroll up the terminal screen as they are installed. Once the bootstrap programs have been installed, they will run automatically and complete the installation of the core programs (FTBP file).
- Once the core FTBP programs are installed, a dialog box will be displayed where you can select any optional program libraries (GUI, ObjectBridge, SUI) and sample programs and data that you want to install.
- If you have previously installed the ViaDuct host programs, the ViaDuct FT proc conflicts with AccuTerm's FT program. If this conflict is detected, the AccuTerm installer will prompt if you want to rename the ViaDuct FT command as "FT.VIA". Answer **y** to rename.

See also:

[Updating the MultiValue Host Programs](#)

[Activating other accounts to use the AccuTerm host programs](#)

4.1.2 Updating the MultiValue Host Programs

To update the MultiValue host programs previously installed on your host, perform the following steps.

- Open a terminal session and connect to your MultiValue host system. Log in to the **ACCUTERM** account (or other account where you have the host programs installed). Make sure you are at a command prompt (TCL, ECL, jsh, etc.)
- While at the command prompt use **MultiValue ? Host Programs ? Update** from the main menu to start the host program update process. Confirm that you want to update the host programs by clicking the **Yes** button.
- The **AccuTerm Host Program Installation Options** dialog box will be displayed. Select the **Reinstall FTBP** option. Click the **OK** button to reinstall the core FTBP programs.
- To update the optional programs and/or sample data files, use **MultiValue ? Host Programs ? Update** from the main menu to restart the host program update process. Confirm that you want to update the host programs by clicking the **Yes** button.
- The **AccuTerm Host Program Installation Options** dialog box will be displayed again. Select any optional program libraries (GUI, ObjectBridge, SUI) and sample programs and data that you

want to install or update.

See also:

[Activating other accounts to use the AccuTerm host programs](#)

4.1.3 Activating Other Accounts to use the AccuTerm Host Programs

After installing the host programs in the **ACCUTERM** (or other) account on your host system, you can easily activate these programs on other accounts. To activate the programs on another account, perform the following steps:

1. Open a terminal session and connect to your MultiValue host system. Log in to the **ACCUTERM** account (or other account where you have the host programs installed). Make sure you are at a command prompt (TCL, ECL, etc.)
2. While at the command prompt use **MultiValue ? Host Programs ? Account Setup** from the main menu to start the host program account setup process. Confirm that you want to activate an account to use the host programs by clicking the **Yes** button.
3. The character-based **FTSETUP** program will start. From the **FTSETUP** menu, select option 1 (enter 1, then press **ENTER**).
4. When prompted for the account name to activate, type the name and press **ENTER**.
5. Repeat for any other accounts that you want to activate.
6. To terminate the **FTSETUP** program, enter **x** at the **FTSETUP** menu, then press **ENTER**.

4.1.4 Downloading Documents to your PC

Use the **FT** utility to send MultiValue items (documents) to your PC. **FT** downloads each item (document) into a separate PC file. Normally, **FT** uses *text* mode, converting attribute-marks to **CR LF** during the download process.

To download using **FT**, enter **FT** at the command prompt. On-line help is provided for each prompt, just enter a question mark and press **ENTER**.

```
(S)end, (R)eceive, (C)onfigure, (O)ptions, (H)elp or (E)xit ?
```

Answer **s** to send, then select the file transfer protocol (answer **C** to modify **FT** [configuration](#) settings; enter **O** to modify **FT** [options](#)):

```
File transfer protocol: (A)SCII or (K)ermit ?
```

Kermit is recommended.

```
Enter source (Pick) file name:
```

Enter the name of the host file which contains the items to be sent.

```
Enter source (Pick) item list:
```

Enter the 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 PC directory name including drive letter if desired. Separate directory names in a full path using back-slashes. UNC-style file names are acceptable.

Enter target (DOS) file name:

Enter the PC file name, including extension. If multiple items are being sent, multiple file names may be specified here. Each host item will be written to a separate PC file. If no file names are specified here, the host item-ID will be used as the PC file name.

When the transfer is complete, you may see a message like:

Converting file to ANSI character set . . .

This indicates that the raw data has been downloaded and is being converted from the host's character set to the ANSI (Windows native) character set.

4.1.5 Uploading Documents from your PC to the Host

Use the FT utility to send PC files to your MultiValue host. Each PC file uploaded is saved as an item in the host file you specify.

To upload using **FT**, enter **FT** at the command prompt and follow the prompts. On-line help is provided for each prompt, just enter a question mark and press ENTER.

(S)end, (R)eceive, (C)onfigure, (O)ptions, (H)elp or (E)xit ?

Answer **R** to receive, then select the file transfer protocol (answer **C** to modify **FT** [configuration](#) settings; enter **O** to modify **FT** [options](#)):

File transfer protocol: (A)SCII or (K)ermit ?

Kermit is recommended.

Enter source (DOS) directory name:

Enter the name of the PC directory which contains the files to be received.

Enter source (DOS) file name:

Enter the name(s) of the PC file(s) to be received. Separate multiple file names with blanks. Wildcard characters (?) and (*) are valid.

Enter target (Pick) file name:

Enter the host file name where the uploaded PC files are to be saved. Each uploaded file is stored as an *item* in this file.

Enter target (Pick) item list:

Enter the item-ID(s) for the uploaded file(s). If multiple files are being uploaded, multiple item-IDs may be specified here. If no item-IDs are specified here, the PC file name will be used as the item-ID.

Before the transfer begins, you may see a message like:

Converting file to host character set . . .

This indicates that the PC file, which uses the ANSI (Windows native) character set is being translated into the host's character set prior to transfer to your host computer.

4.1.6 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
  5. Convert character set..... No
```

Enter parameter number (1-5) to modify:

The setting for any of these parameters may be changed by entering the parameter number.

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
  3. Overwrite existing item..... 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.

Transfer mode

Possible transfer modes are text and binary. Text causes attribute mark characters to be translated to carriage return / line feeds and vice versa.

Preserve file extension

This option affects uploads only. When set to "Yes", and target item-IDs are not specified, the complete DOS file name, including the file extension, is used for the item-ID. When set to "No", the extension is removed.

Overwrite existing item

This option can be used to protect existing files and items from being overwritten. Select "Yes" to allow existing DOS files to be overwritten during downloads (host to PC), or existing items to be overwritten during uploads (PC to host). Select "No" to protect existing files (downloads) or items (uploads).

Large item treatment

For uploads (PC to host), "Truncate" will truncate large files at about 30000 bytes, "Split" will split large files into multiple items of about 30000 bytes each, and "No action" will leave large file unchanged. Note: if you select "No action", be sure your host system does not have an item size limit. Split items use a special item-ID generated by concatenating the original item-ID, an underscore (_), and a sequential number starting with 001. For downloads (host to PC), when "Split" is selected, the process is reversed. That is, multiple items, using the naming scheme described above, are joined together to form a single large DOS file.

Convert character set

If you enable character set conversion, the content of downloaded files is converted from the host's character set to the encoding specified in the session settings *Defaults: Text file encoding*. For uploads, unless the default encoding is set to "no encoding", the actual file contents are examined to determine the file encoding, and the content is converted to the host character set during the transfer.

4.1.7 Downloading Data Files to your PC (export)

Use the File Transfer Wizard or **FTD** utility to download a set of specified (or selected) items from your host database to a PC file. You can specify which fields (attributes) to download. The target PC file format is determined by the extension of the file name you specify.

FTD can also be used to join several host items into a single PC file. This is done by setting the field delimiter to "CrLf", and entering an asterisk at the "Source item list" prompt.

To export data from the host using **FTD**, enter **FTD** at the command prompt, then answer the prompts. On-line help is provided for each prompt, just enter a question mark and press ENTER.

```
(S)end, (R)eceive, (C)onfigure, (O)ptions, (H)elp or (E)xit ?
```

Answer **S** to send, then select the file transfer protocol (answer **C** to modify **FTD** [configuration](#) settings; enter **O** to modify **FTD** [options](#)):

```
File transfer protocol: (A)SCII or (K)ermit ?
```

Kermit is recommended.

```
Enter source (Pick) file name:
```

Enter the name of the host file which contains the data to be exported.

```
Enter source (Pick) item list:
```

Enter the item-IDs of the items (records) you want to export. If you do not enter any item-IDs, any previously selected (SELECT, GET-LIST, etc.) items will be exported; if you enter an asterisk (*), all items in the file will be exported. 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 PC file name including drive letter and directory, if desired. Separate directory names in a full path using back-slashes. UNC-style file names are acceptable.

For certain file types (Excel, Access, etc.), you may need to append a modifier after the file name; see the discussion of [file types](#) for more information.

Attributes to transfer:

Enter the list of fields, or attributes, to be exported. Enter either the attribute number or attribute name as defined in the file dictionary. An asterisk (*) may be entered here to export all attributes including the Item-ID, from all selected items. A plus-sign (+) may be entered to export all attributes *except* the Item-ID.

Generate Header Record (<Y>/N/C):

Answer **Y** or **C** if you want the first record in the PC file to be a "header record". The header record specifies the names of the fields of each data record in the file and may be used as column or field identifiers by certain Windows applications. The **C** option uses the column heading from the dictionary as the header text. Otherwise the dictionary name will be used as the header text. If an asterisk was entered for the attribute list, the header record will contain only an asterisk.

If asterisk or plus-sign was not entered for the attribute list,

Explode MultiValue Fields (Y/<N>):

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.

When the transfer is complete, you may see a message like:

Converting file to Excel file format . . .

This indicates that the raw data has been downloaded and is being formatted in the native file format indicated by the file name you entered.

4.1.8 Uploading Data Files from your PC to the Host (import)

Use the File Transfer Wizard or 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 (answer **C** to modify **FTD configuration** settings; enter **O** to modify **FTD options**):

File transfer protocol: (A)SCII or (K)ermit ?

Kermit is recommended.

Enter source (DOS) file (path) name:

Enter the full PC file name including drive letter and directory, if desired. Separate directory names in a

full path using back-slashes. UNC-style file names are acceptable.

For certain file types (Excel, Access, etc.), you may need to append a modifier after the file name; see the discussion of [file types](#) for more information.

Use Header Record (<Y>/N):

Answer *y* if you want to use the first record in the PC file as a "header record". The header record specifies the names (or numbers) of the attributes to be imported. If the header record consists of a single asterisk, all fields (attributes) in the item are assumed. If you answer *y* to this prompt, the next prompt will be "Enter target (Pick) file name", the other prompts will be bypassed.

Attributes to transfer:

Enter the list of attributes (fields) to be imported. Enter either the attribute number or attribute name as defined in the file dictionary. An asterisk may be entered here to import all attributes (entire record or item). Attribute names or numbers listed here are in the same order as the fields in the DOS file. Enter a dash (-) to skip a field in the DOS file. 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 PC file which are to be considered "header records", and which are to be ignored when importing the file.

Automatically generate target (Pick) IDs (Y/<N>):

Answer *y* to generate 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 host file where the imported data is to be stored.

Before the transfer begins, you may see a message like:

Converting file from Excel format for upload

This indicates that the PC file data is being converted into a delimited ASCII format which can be imported to your MultiValue database prior to transfer to your host computer.

4.1.9 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..... Truncate
  4. Field delimiter..... Tab
  5. Convert character set..... No
```

Enter parameter number (1-5) to modify:

The setting for any of these parameters may be changed by entering the parameter number.

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 fourth configuration parameter to "tab", respond to "Enter parameter..." with **4=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.

Preserve file extension

This option affects FT uploads only. See the [FT options](#) for more information.

Overwrite existing item

This option can be used to protect existing files and items from being overwritten. Select "Yes" to allow existing DOS files to be overwritten during downloads (host to PC), or existing items to be overwritten during uploads (PC to host). Select "No" to protect existing files (downloads) or items (uploads). Set to "Merge" to append or merge data into existing files during downloads.

Large item treatment

This option affects FT operations only. See the [FT options](#) for more information.

Field delimiter

The field delimiter (PC file) may be set to "Tab", "Comma", "CrLf" or any user-defined string. When "Comma" is chosen, fields are enclosed in double-quote marks during downloads (host to PC) and are removed from fields during uploads (PC to host) if present. "CrLf" can only be used for exporting data to the PC (when importing, the field and record delimiters must be different). User-defined field delimiter is entered in hex, at the "Enter parameter..." prompt, as in **4=1A**.

Convert character set

If you enable character set conversion, the content of downloaded files is converted from the host's character set to the encoding specified in the session settings *Defaults: Text file encoding*. For

uploads, unless the default encoding is set to "no encoding", the actual file contents are examined to determine the file encoding, and the content is converted to the host character set during the transfer.

4.1.10 FTD File Types

The default file format used by the **FTD** utility is ASCII delimited text. In this format, records in the PC file are separated by a carriage return / line feed and fields are separated by tabs or commas. If the field delimiter is a comma, then each field is enclosed in quotation marks. Embedded quotation marks are duplicated. This file format is easily imported by many Windows database and spreadsheet programs.

Certain file extensions are recognized by AccuTerm and are assumed to be a particular format. These extensions are:

.accdb	Microsoft Access database
.db2	dBase II
.dbf	dBase III/IV
.htm, .html	HTML table
.mdb	Microsoft Access database
.sym	Lotus Symphony
.wb1	Quattro
.wk1	Lotus v1-v2
.wks	Lotus v3-v5
.xl, .xls	Microsoft Excel
.xlsb, .xlsm, .xlsx	Microsoft Excel 2007

Excel import and export

When exporting and importing from 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 exporting data to Excel, the `range` identifies the upper-left cell of the target worksheet; the number of columns and rows is determined by the data being sent. If a `range` is specified and the Overwrite option is set to "Yes", new data is written beginning at the upper-left cell of the specified range. Cells outside the specified range are not modified. If Overwrite is set to "Merge", you normally do not specify a `range`. In this case, new data is appended to the specified `sheet`. If no `sheet` is specified, a new worksheet is created.

When importing data from Excel, the `range` specifies the entire block of data to be imported. To specify only the upper-left corner of the block (the number of columns and rows determined by the extent of the worksheet), follow the cell with a trailing colon (e.g. B3:).

Access import and export

When exporting and importing from Access databases, the table name must be appended to the file name as:

```
d:\path\file.mdb!table!connect
```

where `d:\path\file.mdb` is the path of the Access database file, `table` is the name of an existing or new table, and `connect` is an optional connection string that may be required to connect to the specified data source. If `Overwrite` is set to "Yes", and the specified `table` exists, then the table is cleared and new data is added. If `Overwrite` is set to "Merge", then new data is appended to an existing table.

4.1.11 Transferring Data Between MultiValue Hosts

The FTPICK utility will transfer specified (or selected) Pick items from one MultiValue computer to another. This transfer can be done in one step if you have two AccuTerm sessions open, connected to each of the MultiValue host computers. AccuTerm will also let you transfer data from one host to another using an intermediate file stored on your PC. This transfer requires two steps: transfer from first host to intermediate file, then transfer from the intermediate file to the second host.

[Direct Host-to-Host \(one step\) transfer](#)

[Host-to-Host transfer using Intermediate PC file \(two step\)](#)

[FTPICK Options and Settings](#)

4.1.11.1 Direct Host-to-Host file transfer

The direct transfer requires two sessions running on AccuTerm - one session connected to the file source host (sending machine), and one session connected to the target host (receiving machine). If you cannot run two sessions, you must use an intermediate PC file to store the data temporarily (see [Host-to-Host transfer using Intermediate PC file](#)).

When using direct transfer mode, both AccuTerm sessions must be running inside the same main AccuTerm window.

To use the **FTPICK** direct transfer method, activate the AccuTerm session connected to the target host, 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 screen. You can cancel the Host-to-Host transfer at any time by pressing the **ESC** key while the target session is active.

Next, activate the AccuTerm session connected to the source host, 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 source host file to be sent:

```
Enter PICK source file name:
```

Enter the name of the source host file which contains the items to be sent.

```
Enter source (Pick) item list:
```

Enter the 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 machine.

```
Enter target PICK file name:
```

Enter the file name on the target 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.

4.1.11.2 Host-to-Host file transfer

If you cannot open one session to each MultiValue host machine, or if you want to send the same file to several target machines, you can save the data in an intermediate PC file. The procedure is similar to the direct method, except that the send must be done first, and a PC file must be specified for both the send and receive steps.

Step 1: Transferring Host items to an intermediate DOS file

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 source file to be sent:

```
Enter PICK source file name:
```

Enter the name of the source file which contains the items to be sent.

```
Enter source (Pick) item list:
```

Enter the 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 **r** to send the file to an intermediate PC file.

Enter intermediate DOS file name:

Enter the full PC file name of the intermediate file including drive letter and directory, if desired. Separate directory names in a full path using back-slashes.

Upon entering the PC file name, the selected items will be transferred to the specified PC file in a special format used by the FTPICK utility.

Step 2: Transferring From an Intermediate File to the Target Host

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 PC file which contains the desired data:

Receive (D)irectly from PICK file or from (I)ntermediate DOS file ?

Answer **r** to retrieve data from an intermediate file.

Enter intermediate DOS file name:

Enter the full PC file name of the intermediate file including drive letter and directory, if desired. Separate directory names in a full path using back-slashes.

Enter target PICK file name:

Enter the file name on the target machine into which the retrieved items will be written.

If the intermediate PC file is in the correct format, the items contained in the file will be transferred and stored into the target host file.

4.1.11.3 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.

4.1.12 Integrating File Transfer Functions into Pick Applications

There are two ways to integrate AccuTerm file transfer functions into your applications: either call one of the supplied file transfer subroutines, described below, or use one of the command-line file transfer programs, passing parameters via STON or DATA statements.

File Transfer Subroutines

There are 7 callable file transfer subroutines that are provided for integrating AccuTerm file transfer functions with your applications:

FTSEND sends one or more items from the host to the PC.

FTRECV retrieves one or more files from the the PC.

FTEXTORT exports a specified or selected set of items from the host database to a PC file. Several file formats are supported including delimited text, Excel worksheet and Access database table.

FTEXTORTDA exports a dynamic array from the host to a PC file. Several file formats are supported including delimited text, Excel worksheet and Access database table.

FTIMPORT imports data from a PC file, worksheet or table, updating one or more items in the host database. The PC file being imported should be a database or table file, such as a spreadsheet, delimited text file, html table or other supported database.

DOSSVC performs three functions (each is optional). First, a "file" is sent to the PC; next a Windows or DOS command is executed (this may be a BATCH file); finally, a file is retrieved from the PC.

FTS performs a simple "file" transfer to or from the PC. The "file" (a dynamic array) to be sent to (or received from) is passed as an argument to the subroutine.

Command-line File Transfer Utilities

As an alternative to calling one of the subroutines, you can use the **FT** and **FTD** command-line 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.

4.1.12.1 FTSEND

The **FTSEND** subroutine sends one or more items from the host to the PC. Each MultiValue *item* becomes a *file* on the PC. The calling syntax is:

```
CALL FTSEND(MODE,HOST.FILE,HOST.IDS,REMOTE.PATH,
            REMOTE.FILES,ITEM.COUNT,BYTE.COUNT,STAT)
```

Input arguments:

MODE	options separated by commas (first letter is sufficient): <ul style="list-style-type: none"> • <u>K</u>ERMIT - use the Kermit file transfer protocol (recommended) • <u>A</u>SCII - use the ASCII file transfer protocol • <u>O</u>VERLAY - overwrites existing files • <u>N</u>O OVERLAY - skips existing files • <u>T</u>EXT - convert AM to CR/LF • <u>B</u>INARY - no AM conversion • <u>U</u>NLIMITED - no item size limit (overrides "large item" config setting) • <u>L</u>IMIT - 32K item size limit (overrides "large item" config setting) • <u>J</u>OIN - joins large item (overrides "large item" config setting) • <u>Y</u> - perform character set translation • <u>Z</u> - no character set translation
HOST.FILE	source file name on host
HOST.IDS	source item list: <ul style="list-style-type: none"> • explicit item-ID list separated by AM • '*' (asterisk) to send all items in the file • null to use an active external select list • (SAVEDLIST to use a saved list
REMOTE.PATH	target directory on the PC
REMOTE.FILES	target file names (used to rename files): <ul style="list-style-type: none"> • explicit file name list separated by AM, corresponding to each item being sent • null to use the source item-ID as the target file name

Output arguments:

ITEM.COUNT	number of items sent
BYTE.COUNT	number of bytes sent
STAT	null indicates success, otherwise an error message

4.1.12.2 FTRECV

The **FTRECV** subroutine retrieves one or more files from the the PC. Each PC *file* becomes a MultiValue *item*. The calling syntax is:

```
CALL FTRECV(MODE,HOST.FILE,HOST.IDS,REMOTE.PATH,
            REMOTE.FILES,ITEM.COUNT,BYTE.COUNT,STAT)
```

Input arguments:

MODE	options separated by commas (first letter is sufficient): <ul style="list-style-type: none"> • <u>K</u>ERMIT - use the Kermit file transfer protocol (recommended) • <u>A</u>SCII - use the ASCII file transfer protocol • <u>O</u>VERLAY - overwrites existing files • <u>N</u>O OVERLAY - skips existing files • <u>T</u>EXT - convert CR/LF to AM • <u>B</u>INARY - no CR/LF conversion • <u>P</u>RESERVE EXT - includes file extension in target item-ID • <u>I</u>GNORE EXT - removes file extension from target item-ID • <u>U</u>NLIMITED - no item size limit (overrides "large item" config setting) • <u>L</u>IMIT - 32K item size limit (overrides "large item" config setting) • <u>J</u>OIN - joins large item (overrides "large item" config setting) • <u>Y</u> - perform character set translation • <u>Z</u> - no character set translation
REMOTE.PATH	source directory on the PC
REMOTE.FILES	source file names, separated with AM; wildcard characters '*' and '?' are allowed.
HOST.FILE	target file name on the host
HOST.IDS	target item list (used to rename files): <ul style="list-style-type: none"> • explicit item-ID list separated by AM, corresponding to each file being received • null to use the source file name as the target item-ID

Output arguments:

ITEM.COUNT	number of items received
BYTE.COUNT	number of bytes received
STAT	null indicates success, otherwise an error message

4.1.12.3 FTEXTORT

The **FTEXTORT** subroutine exports a specified or selected set of items from the host MultiValue database to a PC file. You can specify which fields (attributes) to export. Several file formats, including

Excel worksheet and Access database table, are supported by FTEXTORT. The calling syntax is:

```
CALL FTEXTORT(MODE,HOST.FILE,HOST.IDS,REMOTE.FILE,
ATTRS,ITEM.COUNT,BYTE.COUNT,STAT)
```

Input arguments:

MODE	<p>options separated by commas (first letter is sufficient):</p> <ul style="list-style-type: none"> • <u>K</u>ERMIT - use the Kermit file transfer protocol (recommended) • <u>A</u>SCII - use the ASCII file transfer protocol • <u>O</u>VERLAY - overwrites existing files • <u>N</u>O OVERLAY - skips existing files • <u>M</u>ERGE - appends data to existing file, worksheet or table. If file, worksheet or table does not exist, it is created. • DELIM=TAB - creates a tab-delimited text file • DELIM=COMMA - creates a comma-delimited text file (CSV) • DELIM=CRLF - merges multiple items into a single text file • DELIM=hex - creates a delimited text file with user-defined delimiter • <u>H</u>EADER - first line of target file contains attribute names as column headings • <u>L</u>AABEL - first line of target file contains column headings from file dictionary • <u>E</u>XPLODE - repeats single-valued columns based on number of multi-values in the record • <u>Y</u> - perform character set translation • <u>Z</u> - no character set translation
HOST.FILE	source file name on host
HOST.IDS	<p>source item list:</p> <ul style="list-style-type: none"> • explicit item-ID list separated by AM • '*' (asterisk) to send all items in the file • null to use an active external select list • (SAVEDLIST to use a saved list
REMOTE.FILE	target file name on the PC. See FTD File Types for special file syntax for updating ranges of existing worksheets or specifying Access database tables.
ATTRS	Multi-valued list of attributes to export. Use attribute numbers or dictionary names, or '*' (asterisk) for all attributes, or '+' (plus sign) for all attributes except the item-ID.target file names (used to rename files).

Output arguments:

ITEM.COUNT	number of items exported
------------	--------------------------

BYTE.COUNT	number of bytes exported
STAT	null indicates success, otherwise an error message

Note: it is possible to have partial success, such as when the user clicks the Abort button while a transfer is in progress. Check the ITEM.COUNT & BYTE.COUNT variables to test for partial success.

4.1.12.4 FEXPORTDA

The **FEXPORTDA** subroutine exports a dynamic array from the host to a PC file. The data to be exported is passed as a subroutine argument as a dynamic array. Each attribute is exported as a row in the target file and each value is a column within the row. Several file formats, including Excel worksheet and Access database table, are supported by FEXPORTDA. The calling syntax is:

```
CALL FEXPORTDA(MODE, CONTENTS, ATTRS, REMOTE.FILE,
               ITEM.COUNT, BYTE.COUNT, STAT)
```

Input arguments:

MODE	options separated by commas (first letter is sufficient): <ul style="list-style-type: none"> • <u>K</u>ERMIT - use the Kermit file transfer protocol (recommended) • <u>A</u>SCII - use the ASCII file transfer protocol • <u>Q</u>OVERLAY - overwrites existing files • <u>N</u>O OVERLAY - skips existing files • <u>M</u>ERGE - appends data to existing file, worksheet or table. If file, worksheet or table does not exist, it is created. • DELIM=TAB - creates a tab-delimited text file • DELIM=COMMA - creates a comma-delimited text file (CSV) • DELIM=CRLF - merges multiple items into a single text file • DELIM=hex - creates a delimited text file with user-defined delimiter • <u>H</u>EADER - first line of target file contains attribute names as column headings • <u>Y</u> - perform character set translation • <u>Z</u> - no character set translation
CONTENTS	dynamic array containing data to be exported
ATTRS	multi-valued list of column headings, if HEADER option is specified.
REMOTE.FILE	target file name on the PC. See FTD File Types for special file syntax for updating ranges of existing worksheets or specifying Access database tables.

Output arguments:

ITEM.COUNT	number of items exported
BYTE.COUNT	number of bytes exported
STAT	null indicates success, otherwise an error message

Note: it is possible to have partial success, such as when the user clicks the Abort button while a transfer is in progress. Check the `ITEM.COUNT` & `BYTE.COUNT` variables to test for partial success.

4.1.12.5 FTIMPORT

The **FTIMPORT** subroutine imports data from a PC file, worksheet or table. The PC file being imported should be a database or table file, such as a spreadsheet, delimited text file, html table or other database table supported by AccuTerm. The calling syntax is:

```
CALL FTIMPORT(MODE,REMOTE.FILE,HOST.FILE,ATTRS,HDR.SKIP,
              ID.PREFIX,ID.START,ITEM.COUNT,BYTE.COUNT,STAT)
```

Input arguments:

MODE	<p>options separated by commas (first letter is sufficient):</p> <ul style="list-style-type: none"> • <u>K</u>ERMIT - use the Kermit file transfer protocol (recommended) • <u>A</u>SCII - use the ASCII file transfer protocol • <u>O</u>VERLAY - overwrites existing files • <u>N</u>O OVERLAY - skips existing files • <u>M</u>ERGE - merges data with existing items. Attributes not specified in the ATTRS argument are not updated • DELIM=TAB - imports from a tab-delimited text file • DELIM=COMMA - imports from a comma-delimited text file (CSV) • DELIM=hex - imports from a delimited text file with user-defined delimiter • <u>H</u>EADER - first line of target file contains column headings which are used as dictionary names. ATTRS argument is ignored if this option is specified. If first line of source contains a single asterisk, the imported item is created from all the columns in the source and no data conversions (date, mask, etc) are performed.attribute names as column headings • <u>G</u>ENERATE - generates sequential item-IDs for imported items. The item-ID is formed by concatenating the specified ID.PREFIX to a sequential number starting with ID.START • <u>Y</u> - perform character set translation • <u>Z</u> - no character set translation
REMOTE.FILE	source file on the PC. See FTD File Types for special file syntax for importing ranges of worksheets or specifying Access database tables
HOST.FILE	target file name on the host
ATTRS	Multi-valued list of attributes to import. Use attribute numbers or dictionary names, or '*' (asterisk) for all attributes. Use '-' (minus sign) to skip a column in the source file. Use '0' (zero) for the column that contains the item-ID. If no columns contain item-ID, use the GENERATE option in the MODE argument to automatically generate sequential item-IDs for imported data. This argument is ignored if the HEADER mode option is present.

ID.PREFIX	Prefix for automatically generated sequential item-IDs. This argument is ignored unless the GENERATE mode option is present.
ID.START	Starting number for automatically generated sequential item-IDs. This argument is ignored unless the GENERATE mode option is present.

Output arguments:

ITEM.COUNT	number of items imported
BYTE.COUNT	number of bytes imported
STAT	null indicates success, otherwise an error message

Note: it is possible to have partial success, such as when the user clicks the Abort button while a transfer is in progress. Check the ITEM.COUNT & BYTE.COUNT variables to test for partial success.

4.1.12.6 DOSSVC

The **DOSSVC** subroutine performs three functions (each is optional). First, a "file" is sent from the host to the PC; next a Windows or DOS command is executed (this may be a BATCH file); finally, a file is retrieved from the PC. The data to be sent, PC file name, and Windows / DOS command are passed as arguments to the subroutine, and the received file is passed back to the calling program. The calling syntax is:

```
CALL DOSSVC (CMD, SEND.PATH, SEND.FILE, SEND.DATA,
             GET.PATH, GET.FILE, GET.DATA)
```

Input arguments:

CMD	DOS or Windows command to execute after sending the data to the PC. This argument may be null if no command should be executed.
SEND.PATH	Target directory on the PC where SEND.DATA is saved before executing CMD.
SEND.FILE	Target file name on the PC. If this argument is null, no data is sent to the PC.
SEND.DATA	Contents are stored in the target file on the PC before executing CMD.
GET.PATH	Source directory on the PC where GET.DATA is retrieved from.
GET.FILE	Source file name on the PC where GET.DATA is retrieved from. If this argument is null, no data is retrieved from the PC after executing CMD.

Output arguments:

GET.DATA	Contents of GET.FILE retrieved from the PC after executing CMD.
----------	---

Note: the SEND.DATA and GET.DATA arguments must be different variables!

4.1.12.7 FTS

The **FTS** subroutine performs a simple "file" transfer to or from the PC. The "file" (a dynamic array) to be sent to (or received from) is passed as an argument to the subroutine. The calling syntax is:

```
CALL FTS(MODE, PATH, FILE, ITEM, STAT)
```

Input arguments:

MODE	options separated by commas (first letter is sufficient): <ul style="list-style-type: none">• <u>S</u>END - send file to PC• <u>R</u>ECEIVE - retrieve file from PC• <u>K</u>ERMIT - use the Kermit file transfer protocol (recommended)• <u>A</u>SCII - use the ASCII file transfer protocol• <u>O</u>VERLAY - overwrites existing files• <u>N</u>O OVERLAY - skips existing files• <u>T</u>EXT - convert AM to CR/LF• <u>B</u>INARY - no AM conversion• <u>P</u>RESERVE EXT - includes file extension in target item-ID• <u>I</u>GNORE EXT - removes file extension from target item-ID• <u>X</u>LATE - perform character set translation
PATH	directory on the PC where the item is sent to or retrieved from
FILE	file name on the PC, including file extension, where the item is sent to or retrieved from.
ITEM	body of item to send to PC

Output arguments:

ITEM	body of item retrieved from PC
STAT	null indicates success, otherwise an error message

4.2 phiReport

AccuTerm 8 includes a powerful, user friendly, GUI-based reporting tool called **phiReport**. Developed by 3phi Solutions, **phiReport** is a tool that lets users of MultiValue databases easily create reports in Microsoft Excel.

phiReport Basic is a premium AccuTerm feature, and is available at no charge when an AccuTerm 8 maintenance subscription is active. **PhiReport Professional** supports more advanced features and is available for purchase separately.

phiReport Basic includes:

- Build queries using file dictionary
- Create Excel reports
- Flexible dictionary-based column selection
- Easily specify selection criteria
- Fast preview of report data

phiReport Pro adds these features:

- Email report
- Report security
- Dictionary management using Data Views
- Access to reports from TCL
- Access to Reports API
- Form printing and custom templates

Click [here](#) for a detailed comparison of Basic and Pro features.

See also:

[Installing phiReport](#)
[Activating phiReport in user accounts](#)
[phiReport documentation](#)

4.2.1 Installing phiReport

To install the **phiReport** reporting tool, perform the following steps.

1. **phiReport** is an AccuTerm GUI application, and requires the AccuTerm host programs, including the optional GUI and Object Bridge libraries. If you do not have the AccuTerm host programs installed, see this [topic](#) for instructions to install the host programs. If you do not have the required optional libraries installed, see this [topic](#) for instructions to update the AccuTerm host programs.
*Note: when installing or updating the host programs, either select the **Install all items** option at the top of the selection dialog, or check the **All items needed for phiReport** check-box.*
2. Open a terminal session and connect to your MultiValue host system. Log on to your system administration account (**SYSPROG**, **DM**, **QMSYS**, **JBASEADM**, **UV**, **UD**, etc.) and create a **PHIREPORT** account (you may need to ask your system administrator to create the **PHIREPORT** account for you).
3. Log on to the **ACCUTERM** account.
4. Type **FTSETUP** at the command prompt (TCL, ECL, jsh, etc.) and press ENTER. From the FTSETUP menu, select option 1 (enter 1, then press ENTER).

5. When prompted for the account name to activate, enter **PHIREPORT** and press **ENTER**. This will activate the AccuTerm host programs on the **PHIREPORT** account.
6. Press **ENTER** to exit from the **FTSETUP** program.
7. Log on to the **PHIREPORT** account..
8. Type **PHINSTALL** at the command prompt (TCL, ECL, jsh, etc.) and press **ENTER**. This will complete the installation of **phiReport**.

See also:

[Activating other accounts to use **phiReport**](#)

[Installing the MultiValue Host Programs](#)

[Updating the MultiValue Host Programs](#)

[Activating other accounts to use the AccuTerm host programs](#)

4.2.2 Activating **phiReport** in other accounts

After installing **phiReport** in the **PHIREPORT** (or other) account on your host system, you can easily activate these programs on other accounts. To activate the programs on another account, perform the following steps:

1. **phiReport** is an AccuTerm GUI application, and requires the AccuTerm host programs, including the optional GUI and Object Bridge libraries. If you do not have the AccuTerm host programs installed, see this [topic](#) for instructions to install the host programs. If you do not have the required optional libraries installed, see this [topic](#) for instructions to update the AccuTerm host programs.
2. In order to activate **phiReport** for a use in any account, you first must activate the AccuTerm host programs for that account. See this [topic](#) for instructions to activate the host programs in user accounts.
3. Log in to the account which you intend to use **phiReport**. While at the command prompt (TCL, ECL, jsh, etc.) type **PHIACCOUNTSETUP** and press **ENTER**. This will complete the activation of **phiReport**.

4.3 MultiValue Server

When AccuTerm is connected to a MultiValue host, and the host is running a special server program, **FTSERVER**, installed with the [AccuTerm host programs](#), other Windows applications (client applications) can request and update data from the 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 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 MultiValue database services to client programs, and cannot be used as a normal terminal. A “Server Mode” status 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**” at the TCL command line. To terminate server mode, click the **Cancel** button below the server status message.

See also:

[Configuring the MultiValue Server](#)

[Using the MultiValue Server](#)

[The Server object](#)

4.3.1 Configuring FTSERVER

After installing the [host programs](#), use the **FTSETUP** utility to configure the server. Use **FTSETUP** to give your host a meaningful server name - the server name is shown in the file browser when opening files from the wED editor or GUI designer. You can also enable or disable individual services (read, write, convert, execute) using **FTSETUP**.

4.3.2 Using the MultiValue Server

When AccuTerm is connected to a MultiValue host, and the host is running a special server program, **FTSERVER** (installed when you install the AccuTerm host programs), other Windows applications (client applications) can request and update data from the host database using AccuTerm's [MultiValue Server](#) object.

To access the MultiValue 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 host server name (you can configure the host server name by running the **FTSETUP** program from the ACCUTERM account and selecting menu option 6).

Next, start the client application (Word, Excel, VB, etc.) and open its VBA editor or equivalent tool (e.g., for Excel, select Developer ⇒ Visual Basic). Add a reference to “AccuTerm 8 MultiValue 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 your server object variable as type `object` rather than as `atMVSvr80.Server`.* Next, add a module (Insert ⇒ Module). In your Sub or Function, declare a variable for the server object:

```
Dim MVServerObject As Object
```

Next, use the `CreateObject()` function to create an instance of the server object:

```
Set MVServerObject = CreateObject("atMVSvr80.Server")
```

Use the [Connect](#) method of the server object to establish a connection to the host:

```
If MVServerObject.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.

When you are finished using the server, you can [Disconnect](#) and destroy the server object:

```
MVServerObject.Disconnect  
Set MVServerObject = Nothing
```

If you are going to use the server object multiple times (perhaps from several 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 MVServerObject As Object  
  
Private Function ConnectServer() As Boolean  
    On Error Resume Next  
    If Not MVServerObject Is Nothing Then  
        ConnectServer = True  
    Else  
        Set PickServerObject = CreateObject("atMVSvr80.Server")  
'at8.0  
        If PickServerObject Is Nothing Then  
            Set PickServerObject = CreateObject("atMVSvr71.  
Server") 'at7.1  
            If PickServerObject Is Nothing Then  
                Set PickServerObject = CreateObject  
("atMVSvr7.Server") 'at7.0  
                If PickServerObject Is Nothing Then  
                    Set PickServerObject = CreateObject  
("atPickServer.Server") 'at97/2000/2k2  
                    End If  
                End If  
            End If  
            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 MultiValue host database access would first call the `ConnectServer()` function to make sure that services are available:

```
Public Function MVRead(File As String, ID As String) As String
    If ConnectServer() Then
        MVRead = MVServerObject.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 MVServerObject.LastError Then
    MsgBox "Server Error: " & MVServerObject.LastErrorMessage
End If
```

Some standard errors returned by the FTSERVER program are shown below; other codes may be returned from called MV/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
261	Unable to write item

For further information on using the MultiValue server, a sample Excel worksheet and Visual Basic application are included in the Samples sub-directory.

See also:

[Server object](#)

4.4 Object Bridge

Object Bridge is a Windows Object extension (ActiveX Automation) for MV/BASIC programs. This allows a MV/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: an MV/BASIC interface implemented as a set of callable subroutines, a "remote procedure call" (RPC) conduit (AccuTerm session) and a Windows object manager (installed with AccuTerm 8). The Object Bridge programs are installed (optionally) along with the [AccuTerm host programs](#). If you did not install the Object Bridge programs at the time you installed the host file transfer programs, you can install them by typing LOAD-ACCUTERM-PROGS at TCL in the ACCUTERM account.

Object Bridge allows a MV/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 MV/BASIC subroutines which provide this functionality are described in the following topics.

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 Visual Basic 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

[Initializing the Object Bridge Environment](#)

[Resetting the Object Bridge Environment](#)

[Creating an Object](#)

[Releasing an Object](#)

[Setting Object Properties](#)

[Getting Object Properties](#)

[Invoking Methods \(calling subroutines and functions on an object\)](#)

[Enabling Event Processing](#)

[Polling for Events](#)

[Completing Event Processing](#)

4.4.1 Initializing the Object Bridge Environment

```
CALL ATINITOBJMGR( ERRMSG, OPTS )
```

Call this subroutine before using any other Object Bridge routine. This verifies that a compatible version of AccuTerm 8 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.

4.4.2 Resetting the Object Bridge Environment

```
CALL 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`.

4.4.3 Creating an Object

```
CALL 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.

```
CALL ATGETOBJECT(OBJNAME, CLSID, OBJID, ERRMSG, OPTS)
```

This routine will attempt to get a reference to an existing object. To get a reference to a document object, specify the OBJNAME argument (usually a file name). To get a reference to an arbitrary, specify 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.

4.4.4 Releasing an Object

```
CALL ATRELEASEOBJECT(OBJID, ERRMSG, OPTS)
```

This routine releases the specified object, as well as any objects "derived" from the specified object (OBJID). A derived object may be an object returned by a call to `ATGETPROPERTY` or `ATINVOKEMETHOD` when the return value is itself another object.

4.4.5 Setting Object Properties

```
CALL ATSETPROPERTY(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 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.

4.4.6 Getting Object Properties

```
CALL ATGETPROPERTY(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 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. Releasing the derived object will not release its creator.

4.4.7 Invoking Methods

```
CALL 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.

4.4.8 Enabling Event Processing

```
CALL 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.

4.4.9 Polling for Events

```
CALL 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.

4.4.10 Completing Event Processing

```
CALL 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.

5 Copyright & Trademarks

Copyright Notices

This document and the accompanying software are copyright 2019 Zumasys, Inc. Portions of the software are copyright 1993-2006 Sax Software / Polar Engineering & Consulting, <http://www.kenai.com/vb4tools>.

Freeimage:

The accompanying software uses the Freeimage open source image library. See <http://freeimage.sourceforge.net> for details. Freeimage is used under the Freeimage Public License version 1.0.

DejaVu Fonts:

Fonts are (c) Bitstream (see below). DejaVu changes are in public domain. Glyphs imported from Arev fonts are (c) Tavmjong Bah.

Bitstream Vera Fonts:

Copyright (c) 2003 by Bitstream, Inc. All Rights Reserved. Bitstream Vera is a trademark of Bitstream, Inc.

phiReport is copyright (c) by 3phi Solutions Inc.

Trademark Notices

AccuTerm is a registered trademark of Zumasys, Inc.

Bitstream Vera is a trademark of Bitstream, Inc.

DEC, VT52, VT100, VT220, VT320 and VT420 are trademarks of Digital Equipment Corp.

IBM is a registered trademark of International Business Machines Corp.

Microsoft and Microsoft Windows are registered trademarks of Microsoft Corp.

Pick, D3 and UniData are trademarks of Rocket Software, Inc.

Print Wizard is copyright Rasmussen Software, Inc.

Procomm is a trademark of Datastorm Technologies, Inc.

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.

Other trademarks and registered trademarks appearing in this document are the property of their respective owners.

Index

- . -

.mdb 289
 .xls 289
 .xlsx 289

- 3 -

3phi Solutions 301

- A -

About AccuTerm 31
 Access 289
 AccountName property 104
 ACCUTERM account 279, 281
 AccuTerm constants 201
 AccuTerm Menu Designer 26
 AccuTerm Object 87
 AccuTerm Object Reference 86
 AccuTerm Programming 207
 AccuTerm User Interface 24
 Action property 102
 Activate method 95, 154
 Activating accounts 282
 Activating an evaluation version 23
 ActiveSession property 88
 ActiveX 86
 Adapter script 56
 Add method 166
 AddClone method 167
 AddItem 106
 ADDS 44
 ADDS 4000 236
 ADDS Viewpoint 236
 ADDS Viewpoint Enhanced 217
 Advanced Pick 279
 AIX 279
 Altos 279
 ANSI 44, 53, 54, 58, 67, 68, 71, 178
 AnsiAppCursor property 115, 171
 AnsiAppKeypad property 115, 172
 AnsiAutoprint property 172

Answerback 44
 Answerback property 115, 172
 AppActivate statement 80
 AppClose statement 80
 AppFind function 80
 AppGetActive function 80
 AppGetPosition statement 80
 AppGetState function 81
 AppHide statement 81
 Application menu 63
 Application Mode 44
 Application object 86
 AppList statement 81
 Apply method 170
 AppMaximize statement 81
 AppMinimize statement 81
 AppMove statement 81
 AppRestore statement 82
 AppSetState statement 82
 AppShow statement 82
 AppSize statement 82
 Arrange method 95
 ASCII 58
 ASCII Codes 272
 AsciiDelay property 116, 172
 AsciiEOL property 116, 173
 ATCREATEOBJECT 307
 ATENDEVENT 309
 ATGETPROPERTY 307
 ATINITOBJMGR 306
 ATINVOKEMETHOD 308
 ATRELEASEOBJECT 307
 ATRESETOBJMGR 307
 ATSETEVENT 308
 ATSETPROPERTY 307
 AttributeMask property 115, 171, 173
 Attributes 48, 50
 atwin.ini 274
 atwin71.ini 274
 Authentication 36
 AutoAnswer property 116, 173
 AutoClose property 88, 116
 Auxiliary port 54, 71

- B -

background image 207
 Background picture 51, 116, 117, 173, 174

BackgroundPictureFile property 116, 117, 173, 174
 BackgroundPictureMode property 116, 117, 173, 174
 BackgroundTransparency property 116, 117, 173, 174
 Backspace 44
 Band Properties 63
 Bands 62
 BandType property 100
 Baud property 117, 174
 Baud Rate 41, 42
 Beep 54
 BkspSendsDel property 118, 174
 Blinking 50
 BoldFont property 118, 175
 Border colors 50
 Border style 48, 50
 Break Key 35, 36
 Break method 154
 BreakKeyAction property 118, 175
 Buffer 44
 BytesIn property 118
 BytesOut property 119

- C -

Caché 279
 CallSub method 106
 CapsLock 51
 CapsLock property 119
 Caption property 100, 102, 119
 Capture method 154
 CaptureEnd method 155
 CaptureFileName property 119
 CaptureMode property 119
 CaptureSource property 120
 case inversion 279
 Certificate 40
 Chain statement 82
 Changed property 120
 character encoding 53
 character set 53, 54, 67, 68, 71, 178
 CharacterMapping property 175
 Charset property 120, 176
 Checked property 102
 Cipher 36, 148, 149, 197, 198
 Citrix 17
 Classic Windows 48

Clear method 155
 ClearSelection method 155
 Client certificate 40
 Clipboard Settings 57
 Close method 96, 155
 Col property 121
 Color 48, 50
 Color property 121
 Color Scheme 48
 Color Settings 48, 50
 Colors() property 121, 176
 Cols property 122
 COM 86
 Comm Port 41
 Command() function 83
 Commands
 Private 207
 Common collection 89
 Common property 88
 Configuring AccuTerm 32
 Configuring FTSERVER 303
 Connect method 106
 Connected property 122
 Connection settings 34
 ConnectTimeout property 123, 177
 Console 44, 268
 Context Menu 62
 Control Characters 272
 Control Codes 272
 Controlling AccuTerm with Scripts 78
 Copy method 155
 CopyHistory method 156
 CopyPasteShortcut property 123
 Copyright 310
 CopySelMode property 123, 177
 Count property 99, 100
 Creating an Object 307
 Creating SSH keys 38
 CursorType property 123, 178
 Custom installation 274
 Customize Menu 62
 Customizing the Menu 26
 CustomMouseTable property 89, 90, 96, 97, 124

- D -

D3 69, 278, 303
 Data Bits 41, 42

Data Capture 25
 ending 71
 starting 71
 DataBits property 124, 178
 DataReady event 166
 Debugging a Script 77
 Default Colors 48
 DefaultCaptureDir property 124, 178
 DefaultTextEncoding property 178
 DefaultXferDir property 124, 178
 DefaultXferMode property 124, 179
 DefaultXferOverwrite property 125, 179
 DejaVu font 53
 Delete method 156
 DeleteItem method 107
 Deselect method 156
 Device driver 54
 Device licensing 34
 Device property 125, 179
 Device settings 34
 DeviceLicenseMode property 125, 179
 Dial method 156
 Dialog method 170
 DialStatus property 125
 Direct Host-to-Host file transfer 290
 DisableAppMode property 126, 180
 Disconnect 34
 Disconnect method 107
 Disconnect warning 21
 DockingArea property 100
 DOSSVC 69, 278, 299
 DOSSVC Subroutine 293
 Download 67
 Download method 107, 156
 DSA key 38
 Dtr/Dsr 41, 42
 Duplex 34
 Duplex property 126, 180

- E -

Echo 34
 Edit Menu 26
 Emulate method 157
 Emulation 44
 Enabled property 102
 encoding 53, 54, 67, 68, 71, 178
 Encryption 36

Environment Variable 44
 ErrorMode property 104
 Euro character 53
 Evaluation Version 23
 Excel 289, 301
 Execute DOS command 207
 Execute method 107
 Export 285
 Export method 107
 ExtCols property 126, 180
 Extension event 166
 ExtMode method 127
 ExtRows property 126, 180

- F -

Features of AccuTerm 8 15
 File Download 67
 File Menu 26
 File Transfer 67
 Direct Host-to-Host file transfer 290
 DOS file to Pick item 286
 DOS file to Pick items 283
 DOSSVC Subroutine 293
 FT options 284
 FT utility 69, 278
 FTD options 288
 FTD utility 289
 FTEEXPORT Subroutine 293
 FTEEXPORTDA Subroutine 293
 FTIMPORT Subroutine 293
 FTPICK options 292
 FTPICK utility 290
 FTRECV Subroutine 293
 FTS Subroutine 293
 FTSEND Subroutine 293
 Host file to PC file 285
 Host items to PC file 282
 Indirect Pick-to-Pick transfer 291
 Integration with applications 293
 PC file to Host file 286
 PC file to Host items 283
 Pick file to DOS file 285
 Pick items to DOS files 282
 Pick Server 303
 File transfer protocol
 ASCII 67, 69, 278
 Kermit 67, 69, 278

File transfer protocol
 Xmodem 67
 Ymodem 67
 Zmodem 67
File transfer protocol options 58
File Transfer Settings 58
File transfer wizard 69, 278
File Upload 68
FileExists function 83, 84, 108, 109
FileName property 127
Fingerprint 38
FIPS-140 38
FKeys() property 127, 180
Flow control 41, 42
Folder Settings 33
Font 53
Font Settings 53
FontName property 128, 181
FontSize property 128, 181
FT 69, 278
FT Options and Settings 284
FTD 69, 278
FTD file types 289
FTD Options and Settings 288
FTEEXPORT 69, 278, 295
FTEEXPORT Subroutine 293
FTEEXPORTDA 297
FTIMPORT 69, 278, 298
FTIMPORT Subroutine 293
FTPICK 69, 278
FTPICK Options and Settings 292
FTRECV 69, 278, 295
FTRECV Subroutine 293
FTS 69, 278, 300
FTS Subroutine 293
FTSEND 69, 278, 294
FTSEND Subroutine 293
FTSERVER 59, 303
FTSETUP 303
FTWIZ 69, 278
Fujitsu 279
Full Duplex 34
FuncBarPos property 89, 90
FuncBarStyle property 89, 90
FuncBarVisible property 89, 90
Function key programming 51

- G -

General Settings 32
GetBlock method 157
GetSelection method 157
GetText method 157
Getting Object Properties 307
GmodeEnable property 128, 181
Green Screen 48
GUI 15

- H -

Half Duplex 34
Handshake 41, 42
Handshake property 128, 182
Hangup method 158
Hardware and Software Requirements 17
Hardware Requirements 17
Height property 90, 129
Hide method 96, 158
History 25
History buffer 46
HistoryRows property 129, 182
host character set 53
host mouse reporting 58, 204
Host programs 67, 69, 278, 279, 282, 291, 303, 306
HostName property 130, 183
HostPort property 130, 183
HostTermType property 130, 183
Host-to-Host file transfer 291
hWnd property 131

- I -

Icon 60
Icon property 131
IConv method 108
ID property 131
IDN 35, 36, 149, 199
Image Selection 66
Import 286
Import method 108
Importing SSH keys 38
Initialize method 171
Initializing the Object Bridge Environment 306

Initiate connection 136
 Input method 158
 InputMode property 131
 Installing the MultiValue File Transfer Programs 279
 Installing the Pick File Transfer Programs 279
 Integrating File Transfer Functions into Pick Applications 293
 Integrating Pick and DOS 69, 278
 Internationalized Domain Names 35, 36, 149, 199
 Internet Explorer 36
 Introduction 14
 Invoking Methods 308
 IsConnected method 109
 ISO 8859-1 53
 Item function 83
 Item property 99
 ItemCount function 83
 ItemExists function 109
 Latin-1 53
 LayoutFilename property 90, 96, 97
 Left property 90, 133
 Legacy Colors 48
 Legacy font 53
 LegibleFont property 134, 185
 License 23
 LICENSEACCEPTED 274
 Line Drawing 48, 50
 Line function 83
 LineCount function 83
 Linux 44
 LoadImage method 158
 Locate method 159, 163
 LockBaudRate property 134, 185
 LockFKeys property 134, 186
 LockKeyboard property 134
 Loopback 34

- J -

jBase 69, 278, 279, 303

- K -

Keepalive 35, 36, 150, 200
 KeepaliveInterval property 132, 183
 Kermit 58, 69, 278
 KermitCRC property 132, 184
 KermitEOL property 132, 184
 KermitErrRetry property 132, 184
 KermitInitRetry property 132, 184
 KermitQuote property 133, 185
 KermitRept property 133, 185
 KermitTimeout property 133, 185
 Key Fingerprint 38
 Key Management 38
 Key Types 51
 Keyboard Commands 25
 Keyboard Settings 51

- L -

Language Extensions 79
 LargeIcons property 90
 LastError property 105
 LastErrorMessage property 105

- M -

Macro Recorder 74
 Main Menu 24, 62, 63
 maintenance 21, 34, 44, 301
 MapUpperFKeys property 134, 186
 Master menu 62, 63
 Master menu override 63
 MDI 20
 MDName property 105
 Mentor 279
 Menu 24, 26
 Menu Bar 62, 63
 Menu Designer 26, 62
 Menu file name 63
 Menu icon 66
 Menu Object 98, 103
 Menu property 91, 135
 Menu Scope 62
 Menu Structure 62
 Menu Tool 64
 menu.ini 274
 messengerows property 135, 186
 Microsoft Excel 301
 MM-MON 268
 MnuBand Object 99
 MnuBands property 99
 MnuTool Object 101
 MnuTools property 101

Modem 42
 Modem settings 42
 Modern Windows 48
 Monochrome 48
 Mouse 46, 89, 90, 96, 97
 host control of 207
 mouse pattern file 58, 204
 mouse pattern matching 58, 204
 Mouse Settings 58
 Mouse Wheel 46
 MouseEnable property 135, 186
 MouseTableAdd method 159
 MouseTableLoad method 159
 MouseTableReset method 159
 Move method 96, 159
 Multiple Document Interface 20
 MultiValue 59, 69, 278, 279, 281, 301, 303
 MultiValue Database 303
 MultiValue File Transfer Programs 279, 281
 MultiValue Host Programs 282
 MultiValue Server 303
 mvBase 279
 mvEnterprise 279

- N -

Nailed Port 44
 Name property 101
 network outage 21, 44
 New Features 18
 NoAutoWrap property 135, 187
 NoCloseWarning property 91
 Normal installation 17
 NormCols property 136, 187
 NormMode method 136
 NormRows property 136, 187
 Northgate 279

- O -

Object hierarchy 86
 Object Reference 86
 AccuTerm object 87
 Creating new session 77
 Events 76
 Menu object 98, 103
 Methods 76

MnuBand object 99
 MnuTool object 101
 Predefined session objects 77
 Predefined sessions objects 77
 Properties 76
 Session object 77, 110
 Sessions collection 166
 Settings object 167
 ObjectBridge 306
 OConv method 109
 OLE 86, 306
 OnLine property 136
 OpenFileName function 84
 OpenInsight 279
 OpenLayout method 96, 97
 OpenSSH 38
 Orientation 141, 145, 191, 195
 Output method 160
 OverrideModemConfig property 136, 187

- P -

Page property 136
 Pages property 137, 187
 Palette 50
 Palette() property 137, 187
 paper size 141, 145, 192, 195
 Paper source 142, 146, 192, 195
 Parity 41, 42
 Parity property 138, 188
 Password 36
 Paste method 160
 PasteAllowCurlyQuotes property 138, 189
 PasteEOFChar property 138, 189
 PasteEOFMode property 138, 189
 PasteEOLChar property 139, 189
 PasteEOLMode property 139, 189
 Pause 25
 Pause statement 84
 PCL 56
 PC-MON 268
 Personal menu 62, 63
 phiReport 301, 302
 PhoneBookName property 91
 Pick 69, 278, 303
 Pick Database 303
 Pick File Transfer and Server Programs 69, 278
 Pick File Transfer Programs 279, 281

Pick File Transfer Programs 279, 281
 installation 69, 278
 Pick PC Console 268
 Pick PC Monitor 44
 Pick Server 303
 Pick-to-Pick file transfer 291
 PlayMidi method 160
 PlayWave method 160
 Popup Menu 62, 63
 Port property 139, 190
 Portable installation 17
 Power 95 279
 Pre-configured installation package 274
 Print adapter 56
 Print adapter script 54
 Print adapters 144, 147, 194, 196
 Print Screen 25
 Print Wizard 54, 144, 147, 194, 196
 Printer driver 54
 Printer paper size 141, 145, 192, 195
 Printer Settings 54
 PrinterClose method 160
 PrinterColorMode property 140, 190
 PrinterFontBold property 140, 190
 PrinterFontItalic property 140, 191
 PrinterFontName property 140, 191
 PrinterFontSize property 140, 191
 PrinterMode property 141
 PrinterName property 141, 147, 196
 PrinterOff method 161
 PrinterOn method 161
 PrinterOrientation property 141, 191
 PrinterPaperSize property 141, 192
 PrinterPaperSource property 142, 192
 PrinterTimeout property 143, 193
 PrintJobEject property 143, 193
 PrintScreen method 161
 PrintScreenBackground property 143, 193
 PrintScreenEject property 143, 194
 PrintWizard 56
 Prism 44
 Private Commands 207
 Private Key 36
 Procomm 236
 Product activation 23
 ProductLicenseType property 91
 ProductName property 91
 ProductRelease property 92

ProductSerialNumber property 92
 Programming AccuTerm 207
 ProtectAttr property 144, 194
 Protected Attribute 46
 PTERM 279
 Public Key 36
 Punycode 35, 36, 149, 199
 putty 38
 puttygen 38

- Q -

QM 69, 278, 279, 303
 Quiet installation 274

- R -

R83 279
 R91 279
 Random function 84
 ReadItem method 109
 Readnext method 109
 ReadText method 161
 Reality 279
 Recent file list 33
 RecentListSize property 92
 Reconnect 21
 RegisteredCompany property 92
 RegisteredLocation property 92
 RegisteredUser property 93
 Releasing an Object 307
 reports 301
 Reset method 161
 ResetComm method 162
 ResetTerm method 162
 Resize method 97, 162
 Resumable 44
 Resume 21, 25, 34
 Revelation 279
 ReZume 21, 34, 44
 Row property 121
 Rows property 122
 RSA key 38
 Rts/Cts 41, 42
 Running a Script 76
 Running AccuTerm 8 20

- S -

- Save method 162
- SaveAs method 162
- SaveFileName function 84
- SaveLayout method 97
- ScaleFont property 144, 194
- Scaling 51
- SCO 44
- Screen print 54
- Screen Settings 46
- Screen Size 46
- ScreenPrintAdapterScript property 144, 194
- ScreenPrintMode property 145, 195
- ScreenPrintOrientation property 145, 195
- ScreenPrintPaperSize property 145, 195
- ScreenPrintPaperSource property 146, 195
- Script 79
 - Chain statement 76
 - closing 75
 - controlling AccuTerm 76
 - creating 75
 - debugging 77
 - editing 75
 - loading 75
 - Pause statement 76
 - printing 75
 - running 76
 - running from command line 76
 - running from function key 76
 - running from host computer 76
 - running from menu or toolbar 76
 - saved in layout file 76
 - saving 75
- Script Language Extensions 79
- Scripting 75, 86
- Scripting language 15
- ScrMode property 146, 196
- Scrollbar 25, 46
- Scrollbar 46
- ScrollHistory method 163
- ScrollMode property 146, 196
- Secure Shell 36
- Select method 163
- Selection 26
- Selection property 146
- Sending a DOS File to a Pick File 286
- Sending a Pick File to a DOS File 285
- Sending DOS Files to Pick Items 283
- Sending MultiValue Items to DOS Files 282
- Sequoia 279
- Serial Port 41
- Serial port settings 41
- Server 303
- Server validation 40
- ServerID property 105
- ServerName property 105
- Session 62
- Session Bar 62, 63
- Session icon 46, 60
- Session Object 110
- Session Tabs 24
- Session title 46, 60
- SessionBarVisible property 93
- Sessions Collection 166
- Sessions property 93
- Sessions.Add method 166
- Sessions.AddClone method 167
- Sessions.Count method 167
- SetBlock method 163
- SetExtension method 163
- SetSelection method 164
- SetText method 164
- Setting Object Properties 307
- Settings
 - Attribute Colors 50
 - Background 51
 - Colors 48, 50
 - File Transfer 58
 - Font 53
 - General 32
 - Keyboard 51
 - Mouse 58
 - Palette 50
 - Printer 54
 - Public Key 38
 - Secure Shell 38
 - Sound 54
 - SSL/Telnet 40
 - Telnet 40
 - Terminal Type 44
- Settings method 144
- Settings Object 167
- SettingsDialog method 97
- setup.ini 274

Shell function 84
 Shortcut 20
 Show method 98, 164
 ShowErrs property 147
 Silent installation 274
 SingleInstance property 93
 Slave port 54, 71
 Slave printer 54, 71
 SlavePrintAdapterScript property 147, 196
 SlavePrinterName property 141, 147, 196
 SlavePrintMode property 147, 197
 Sleep statement 85
 Software License Agreement 274
 Software Requirements 17
 Sound property 148, 197
 Sounds 54
 Speaker 54
 Splash screen 60
 SSH 36, 130, 182, 183
 SSH Cipher 149, 198
 SSHAuth property 148, 197
 SSHCipher property 148, 197
 SSHKey property 148, 198
 SSHUseOnlyPreferredCipher property 149, 198
 SSHVersion property 149, 198
 SSL 40, 120, 129, 176, 182
 SSL/Telnet 40
 Standalone installation 17
 Start capturing 71
 Status Bar 24, 62, 63
 Status bar icon 66
 Status Lines 44
 StatusLineVisible property 94
 Stop Bits 41, 42
 StopBits property 149, 198
 Strip8th property 149, 199
 Sub-Menu 64
 System Colors 48
 System Requirements 17

- T -

TCL Rules 59
 TCP/IP 35, 36, 40
 TcpIpn property 149, 199
 TcpNoDelay Property 150, 199
 Technical support 73
 Tektronix 44

Tektronix Graphics 46
 Telnet 35, 40
 Telnet settings 35, 40
 TelnetBinary property 150, 199
 TelnetBypass property 150, 199
 TelnetKeepalivesNOP property 150, 200
 Terminal Emulation 44
 Terminal Server 17
 Terminal Services 17
 Terminal Type 35, 36, 44
 Terminal Type Settings 44
 Terminate connection 136
 Terminate method 98, 164
 TermType property 151, 200
 Text file encoding 58
 The ScreenBlock object 103
 Themes 48
 Tile 24
 Timeout 35, 36, 40
 Title Bar 24
 TitleFormat property 94
 TLS 40
 Tool Properties 64
 Toolbar 24, 26, 62, 63
 Toolbar icon 66
 ToolbarVisible property 94
 ToolID property 103
 Tools 62, 64
 Tools Menu 26
 ToolTipText property 103
 Top property 94, 152
 TrackKeyboardState property 94
 Trademark 310
 Transfer mode 58
 Transparency 51
 Type library 86

- U -

U2 69, 278, 303
 Ultimate 279
 Unattended installation 274
 Underline 50
 Unicode 15, 53, 54, 58, 67, 68, 71, 106, 178
 UnicodeDelimiters property 106
 UniData 279
 UniVerse 279
 UniVision 279

UnloadImage method 165
UnlockItem method 110
Updating the MultiValue File Transfer Programs 281
Updating the Pick File Transfer Programs 281
Upgrade 18
Upload 67
Upload method 110, 165
UserName property 106
Using the FT Utility 282, 283
Using the FTD Utility 285, 286
Using the FTPICK Utility 290
Using the MultiValue Server 303
Using the Pick Server 303
UTF-8 15, 53, 54, 58, 67, 68, 71, 178

- V -

VARFILES 274
VBA 15, 75
ViaDuct 279
Video Pages 46
View Menu 26
Viewpoint 44, 236
Viewpoint Enhanced 217
Visible property 95, 101, 103, 152
Visual attributes 48, 50
Visual Basic for Applications 75
Visual Styles 48, 50

- W -

WaitFor method 165
wallpaper 116, 117, 173, 174, 207
WED 15
Welcome screen 20
Width property 95, 152
Window 24
Window Menu 26
Windows Terminal Server 17
WindowState property 95, 152
Wizard 26, 69, 278, 282, 283, 285, 286
Word function 85
WordCount function 85
WriteItem method 110
WriteText method 165
Wyse 44
Wyse 50 217

Wyse 60 217

- X -

XferBytes property 153
XferFiles property 153
XferStatus property 153
XModem 58
XmodemTimeout property 153, 201
Xon/Xoff 41, 42

- Y -

YModem 58
YmodemTimeout property 154, 201

- Z -

ZModem 58
ZmodemTimeout property 154, 201
Zumasys company information 73