AccuTerm 7 Release Notes

Release 7.0d (06-20-2011)

AccuTerm 7 is designed to run on Windows XP SP2 and above. It will not run on Windows 2000 or any Windows 9x or ME versions.

Upgrading from AccuTerm 2K2 to AccuTerm 7

AccuTerm 7 should be installed in a separate directory from other versions of AccuTerm. Unlike prior releases, AccuTerm 7 does not share any common components with prior versions of AccuTerm (at least none in the program directory.) The default installation directory is C:\Program Files\Atwin70. The atwin.ini file has been renamed atwin70.ini and the application data directory has moved to C:\Users\username\AppData\Roaming\Asent\atwin70. This is where private menu files, the dialing directory and ssh key files are stored.

Although the host programs from AccuTerm 2K2 are upward compatible with AccuTerm 7, we recommend installing the new host programs from AccuTerm 7. These should be backward compatible with any previous version of AccuTerm. If you use AccuTerm GUI, you definitely should install the new GUIBP library because there are several new subroutines included, and new constants defined in ATGUIEQUATES for new events, properties and methods.

Although you can run LOAD-ACCUTERM-PROGS from the ACCUTERM account on the host to install the updated programs, you can now use **Update Host Program** from the **MultiValue** menu. Be sure to select the **Reinstall FTBP** first, then run the update again and reinstall the other desired programs (especially GUIBP).

Product Activation

The product activation has been updated in AccuTerm 7. The product can now be activated at any time without requiring a re-install. In the **Help** menu, there is a new selection: **Enter Activation Code**.

The Activation form has a check box: **Send registration details to AccuSoft over the Internet**. We recommend checking this box to register your copy of AccuTerm when it is activated. This will aid us in helping you in the event that you lose your activation code or have other support issues.

Importing your Settings

When you run AccuTerm 7 for the first time, it checks if there are settings from older versions of AccuTerm that can be imported into AccuTerm 7. This is a one-time offer. It will display a list of the various settings that can be imported and you can choose which ones to import.

The session configuration files, layout files and script files can be used without any changes. If you update the settings and save the file from AccuTerm 7, you might not be able to use it in AccuTerm 2K2, as some settings are stored using different keys than before. If in doubt, make a backup copy of the file, or use Save As to save using a new name.

New Features - Main Program

The menu, toolbar, status bar and session bar have been completely revamped. The old session bar has been replaced with a new tab set. You can right-click on a tab to display a context menu with options to activate, maximize, close or *clone* the session. There is a drop-down arrow at the right edge of the tab bar to select sessions that are not currently displayed on the bar.

The Settings dialog box has been completely re-designed. The new interface uses a tree to select the category of setting to display in the main pane of the dialog. Using the tree interface has made the settings more logical.

With AccuTerm 7, you can select the printer assigned for screen printing and slave printing separately. Each function has its own independent set of properties.

Network features in AccuTerm 7 have been improved. IPV6 is now supported, as is "Internationalized Domain Names" (punycode). SSH has an additional authentication protocol: Keyboard Interactive. This protocol is similar to Password, but is host-driven. It is in common use on many Linux versions. A main difference between Keyboard Interactive and Password, is that Keyboard Interactive supports password expiration and changing (if supported by the host).

Since a large part of AccuTerm's market is the MultiValue community, those functions where are peculiar to MultiValue users have been consolidated into a new **MultiValue** menu group in the main AccuTerm menu. These features include the GUI Designer, wED Editor and File Transfer Wizard. There are sub-menus for installing, updating and configuring the host programs.

There is now an option in FTD and FTEXPORT to use column headings instead of dictionary names in the heading row when sending data from the host to Windows. A new FTEXPORTDA subroutine has been added to the library to export a dynamic array passed as an argument instead of data from a file.

New Escape Sequences (AccuTerm Programming)

Some new private escape sequences added to AccuTerm 7:

ESC STX p *mode* CR - selects the printer mode:
mode = 0 - turns off auto or transparent print
mode = 1 - turns on auto print
mode = 2 - turns on transparent print
mode = X - turns off auto or transparent print and closes the print job immediately

ESC STX yj, *name* CR - queries status of a stored screen block. Sends a 0 or 1, followed by a CR, indicating if a block of the specified name exists in AccuTerm's screen block memory. Screen blocks are stored using ESC STX jS,... documented in the 2K2 Programmers Guide.

ESC STX iB, *filename* , *scalemode* , *transparency* 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.

transparency - a percentage - 100% means that the picture is the only thing seen in the background. 0% means that the screen background color is visible and the picture is invisible. Any percentage in between blends the picture with the screen background color.

To clear the background image, simply use ESC STX iB CR.

ESC STX R *script* CR - this is a variant of the old ESC STX P command used to run a script. The difference with this command is that the script is synchronized with the terminal. This basically sets the session 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. This has the effect of buffering any data or escape sequences received while the script is running. This is useful when changing Session properties such as screen sizes, terminal type, etc.

Previously, we recommended adding a final Output in scripts that required synchronization. This required a corresponding INPUT statement in the host program. However, this technique is not compatible with typeahead on the host. The new escape sequence solves this problem.

ESC STX w and ESC STX m - these existing escape sequences can be used to play most types of audio media files, including mp3 files.

ESC STX iL,... - this existing escape sequence is used to display images on the text screen. It has been updated to use the FreeImage Open Source image library, so more image formats are now possible.

AccuTerm Object Model Changes

Server object:
The name of the server object provided by AccuTerm 7 has been changed from "atPickServer.Server" to "atMVServer.Server". You need to use the new name in your CreateObject() function to access the AccuTerm 7 version of the MultiValue server. If you are using a reference to the "AccuTerm 2K2 MultiValue Server" type library, you need to change the reference to "AccuTerm 7 MultiValue Server". To make your code version-independent, declare variables that reference the Server object as Object instead of an explicit type. Attempt to create the server object using the new AccuTerm 7 name, and if an error occurs, try using the AccuTerm 2K2 name instead.

AccuTerm object:

- MenuFileName property has been removed - this property has been obsolete since AccuTerm 2K2.
- TelnetHostsFile property is obsolete and has been removed.

Session object:

- FKeys property and FKey object: the FKeys property returns an FKeys collection which acts like the FKeys() array property from previous AccuTerm versions so existing scripts should continue to work. Instead of containing strings, elements in the FKeys collection are FKey objects. The default property of the FKey object is the content of the key, which is a string.
- BackgroundPictureFile property has been added to support background pictures.
- BackgroundPictureMode property has been added to support background pictures.
- BackgroundTransparency property has been added to support background pictures.
- ScreenPrinterName, ScreenPrintPaperSize, ScreenPrintPaperSource, ScreenPrintOrientation properties have been added so that screen & slave printers can be configured independently.
- ScreenPrintAdapterScript property has been added to support use of a "print adapter script" to process screen printing.
- SlavePrinterName property has been added as a synonym for PrinterName property.
- SlavePrintAdapterScript property has been added to support use of a "print adapter script" to process slave print jobs.
- DeviceLicenseMode property has been added to supersede the old U2DeviceLicensing property (old property is still valid).
- BreakKeyAction property has been added to supersede the old TelnetAltBreak and SSHBreakCharacter properties (old properties are still valid).
- KeepaliveInterval property has been added and supersedes the old TelnetKeepalive and SSHKeepalive properties (which remain valid). The new property allows the interval to be specified which whas not possible using the old properties.
- TcpNoDelay property has been added and supersedes the old TelnetNoDelay and SSHNoDelay properties.

TcpIdn property has been added to support "Internationalized Domain Names" or "punycode" for host names.

TelnetKeepaliveIsNOP property has been added to select the telnet NOP command as a keepalive message (the default is the telnet DMARK command).

SSHUseOnlyPreferredCipher property has been added.

OnLine property has been added. Set this property to True to connect or False to disconnect.

CopySelMode property has been added to support new "remove selection after copy" feature.

GetSelection method has been added to return coordinates of selection (instead of selected text returned by the Selection property).

Copy method has been changed so that the last line copied to the clipboard does not have a terminating CR/LF appended.

Server, MenuFileName, DDEResult, ScrAutoFwd properties are obsolete and have been removed.

TelnetEcho, TelnetAltBreak, TelnetNoDelay, TelnetKeepAlive, SSHNoDelay, SSHKeepalive, SSHBreakCharacter and U2DeviceLicensing properties have been superseded by new properties listed above. These properties continue to function, however their use is discouraged.

Settings object:

Generally, the same properties that were added, removed or superseded in the Session object have been similarly updated in the Settings object.

New Features - GUI

The following sections document the new features that available in AccuTerm 7 GUI. Note that there are still a significant number of items on the "wish list", so if your desired "wish" did not make the cut for this release, don't give up hope - it might make a future release. Generally, new features that were requested by multiple users are the ones that have been included, although, not all of these were feasible at this time.

One of the main goals of AccuTerm 7 is to bring the user interface style up to the level of modern Windows versions. Every intent has been made to render the GUI objects according to the style used by Windows itself. Because of this, you are not guaranteed that colors or sizes will be 100% consistent between Windows versions, or even between the same version with different themes, monitor sizes and screen resolutions. This might require a bit more flexibility by developers to ensure their applications look pleasant on a wide range of target systems. There is a system-wide setting in the atguisvr.ini file in the program directory. Add the item DisableThemes=True to the [Behavior] section of this file. You can create the file with Notepad if it does not exist.

Most AccuTerm GUI objects have a "Border Style" property, with settings for None, Flat and 3D. The 3D style has been renamed as "Normal". When using themes (the default) on Windows XP and later, the Normal border style will be rendered using the appropriate Windows border style for the particular object. With Windows XP and above, most borders are drawn using a flat style with colors compatible with the current Windows theme.

AccuTerm 7 GUI has a "type ahead" feature, which was first introduced in AccuTerm 2K2 release 5.3c-sp2. This feature buffers keyboard and mouse events while the host is processing events. So even though the user might not see the effect of a keystroke or click immediately, the short delay is typically barely noticeable and the GUI program appears to be more responsive than previous versions without the type ahead feature. There are several actions that clear any buffered events from the type ahead buffer: calling ATGUIACTIVATE, calling ATGUISUSPEND or ATGUIRESUME and switching to a different Windows application.

New Controls:

Timer

ATGUICREATETIMER(APPID, FORMID, CTLID, STYLE, TIMEOUT, GUIERRORS, GUISTATE)

The ATGUICREATETIMER routine creates a timer control that fires a Timer (GETIMER) event after a specified interval. The timer may be one-shot by specifying STYLE = 0 or repeating by specifying STYLE = 1. For one-shot timers, you can restart the timer by setting the Enabled (GPENABLED) property to True (1). For repeating timers, setting Enabled (GPENABLED) to False (0) will disable the timer. Specify TIMEOUT in milliseconds. The TIMEOUT value can be changed at any time by setting the Timeout (GPTIMEOUT) property. Using the timer control may be more convenient than calling ATGUICHECKEVENT with a timeout because the interval can be controlled more precisely. Multiple timers are supported.

HTML Viewer

ATGUICREATEBROWSER(APPID, FORMID, CTLID, PARENTID, EVENTMASK, URL, LEFT, TOP, WIDTH, HEIGHT, GUIERRORS, GUISTATE)

The ATGUICREATEBROWSER routine creates an HTML Viewer control, which is currently implemented using a component version of Internet Explorer. *It is suggested that you used this control to display simple HTML content, rather than rely on full browser capabilities - future version of the GUI runtime may use a different implementation which may not have all the capabilities of Internet Explorer.*

The HTML Viewer control has a Value (GPVALUE) property which contains the URL you wish to display in the control, as well as a Content (GPCONTENT) property which allows you to load HTML-formatted content for display in the control. The control will fire a Click (GECLICK) event if the user clicks a "topic" link. You can use this event to reload the Content based on the target topic.

For more information, see the ATGUIHELP topic in the online help.

Status Bar

ATGUICREATESTATUSBAR(APPID, FORMID, CTLID, STYLE, ITEMS, GUIERRORS, GUISTATE)

The ATGUICREATESTATUSBAR routine creates a status bar that displays at the bottom edge of a form. A status bar consists of a series of one or more panels and is typically used to display the current status of the form, indicating various status information. Each "panel" is described by a value in the ITEMS argument as follows:

item<1,1,1> =status bar panel ID
item<1,1,2> =unused (must be null)
item<1,1,3> = panel caption
item<1,1,4> = panel icon (picture filename)
item<1,1,5> = enabled
item<1,1,6> = visible
item<1,1,7> = unused (must be null)
item<1,1,8> = unused (must be null)
item<1,1,9> = tooltip
item<1,1,10> = panel size (0=auto, -1=spring, else actual size)
item<1,1,11> = panel alignment (0=left, 1=right, 2=center)

Note: when a panel size is set to -1 (spring), the panel size is adjusted to fill up the remaining space on the status bar. If multiple panels are set to "spring", the extra space is divided among them.

The status bar supports Click (GECLICK), DoubleClick (GEDBLCLICK) and Context (GECONTEXT) events. Panels using built-in IDs do not respond to the Click event (*Caps, *Num, *Scroll, *Date, *Time, *Timedate).

Enhanced Controls

Form

drop (drag not supported)

Label

transparent background (GPTRANSPARENT)
FontUnderline (GPFONTUNDERLINE)
data field style (2=single line data field, 3=multi-line data field) eliminates need to use '&&' when you need to display the '&' character in a label
drag & drop

Edit

New Style property: 0 = single-line, 1=multiline (word-wrap enabled), 2=password, 3=multiline (word-wrap disabled), 4 = up/down button

FontUnderline (GPFONTUNDERLINE)

drag & drop (source and/or target character position is reported in GEDRAGDROP event)

SelRange property returns SelStart in propval<1,1> and SelLength in propval<1,2> (GPSELRANGE)

The edit control GECHANGE event now includes the new content in GUIARGS<1>

The new style 3 multiline with word-wrap disabled displays both horizontal and vertical scroll bars.

The new style 4 up/down buttons is a single-line edit control, with a small "spinner" button at the right edge. Clicking the up or down spinner button will fire a GEBTNCLICK event, with GUIARGS<1,1> = 0 when the "down" button is clicked and GUIARGS<1,1> = 1 when the "up" button is clicked. GUIARGS<1,2> contains the current value.

Command Button

new Toolbar button style (does not get focus)

change to Cancel button style - the Cancel button bypasses "required" & local data type checking.

Icon, IconSize & IconAlign properties display proper icon on button (GPICON, GPICONSIZE, GPICONALIGN)

FontUnderline (GPFONTUNDERLINE)

custom Background & Foreground colors are now supported

drag & drop

The command button has a new event, GESTATUS, which is fired when a mouse button is pressed or released. In this way, an application can perform some action while the button is held down, perhaps incrementing some value at a periodic rate. The GESTATUS event arguments are:

GUIARGS<1,1,1> = button (1 = left, 2 = right, 3 = middle)
GUIARGS<1,1,2> = state of Shift, Ctrl & Alt keys
GUIARGS<1,1,3> = 1 if button is down (pressed), 0 if button is up (released)
GUIARGS<1,1,4> = X position of mouse, relative to the command button's position
GUIARGS<1,1,5> = Y position of mouse, relative to the command button's position

Option

transparent background (GPTRANSPARENT)

FontUnderline (GPFONTUNDERLINE)

ReadOnly property supported (GPREADONLY)

drag & drop (source and/or target button index is reported in GEDRAGDROP event)

Checkbox

transparent background (GPTRANSPARENT)
left (0) or right (1) alignment (GPALIGN)
FontUnderline (GPFONTUNDERLINE)
ReadOnly property supported (GPREADONLY)
drag & drop (GPDRAGID, GPDROPIDS)

List box

new "checkbox" style supports multi-select using checkboxes instead of ctrl+click

New Style property: 0 = single selection, 1=multiple selection, 2= drop-down list, 3=checked multiple selection (GPSTYLE)

first column can display an icon per-item (GPICON)

FontUnderline (GPFONTUNDERLINE)

ReadOnly property supported (GPREADONLY)

new ColDataType property used to indicate data type when sorting content using ATGUISORT (GPCOLDATATYPE)

drag & drop

new search algorithm: using the keyboard to search a list for a particular item is now consistent between the list box, combo box and grid (dropdown list & dropdown combo columns). Previously, the grid employed an incremental search and the list and combo box employed a simple first-letter search. The incremental search is more powerful: each key typed is concatenated to a hidden string that is used to search the list. When a key causes the hidden string to not match any items, the string is reset to the last key entered and the search is repeated. For example, if a list contains items 'car', 'cat' and 'dog', typing 'c' would select 'car', typing 'a' would keep the same selection. Typing 't' would change the selection to 'cat'. Typing 'd' would change the selection to 'dog'.

Combo box

first column of dropdown list can display an icon per-item (GPICON)

return index of selected item using Row (GPROW) property

FontUnderline (GPFONTUNDERLINE)

ReadOnly property supported (GPREADONLY)

new ColDataType property used to indicate data type when sorting content using ATGUISORT (GPCOLDATATYPE)

drag & drop

new search algorithm (see List box above for details)

Picture box

additional picture file formats are supported: BMP, JPEG, GIF, PCD, PCX, PICT, PNG, PSD, TARGA, TIFF, WBMP, XBM, XPM and Windows Metafile

drag & drop

Frame

transparent background (GPTRANSPARENT)

added support for caption-less frames, which do not display a caption, and provide slightly more space within the frame vertically. This feature is enabled by setting GPSTYLE to 1. A new ATGUICREATEFRAME2 subroutine has been added to the library which includes the STYLE argument.

FontUnderline (GPFONTUNDERLINE)

drag & drop

added Click, DblClick, Context events

Grid

Word wrap (GPWORDWRAP)

column-level tab stop overrides default cell navigation (GPCOLTABSTOP)

can select, cut, copy & paste multiple cells (GPSELSTART, GPSELLENGTH, GPSELRANGE)

new column type displays icon + label

new column type, like ellipsis type, displays label + button. Use the GPICON property to set the button icon. Clicking the button fires a GEBTNCLICK event.

ellipsis column type fires GEBTNCLICK event. This is the same event as GEELLIPSIS (same value in ATGUIEQUATES) just a new name. This may cause some confusion in the code generator in the GUI Designer.

new FocusStyle property (GPFOCUSSTYLE) determines the style of focus indicator used to highlight the active cell: 0 = no focus indicator, 1 = thin dotted line, 2 = thick dotted line, 3 = heavy solid line.

new PasteMode property (GPPASTEMODE) determines whether multi-cell paste is allowed (e.g. pasting a block from an Excel worksheet). The default is single-cell paste.

FontUnderline (GPFONTUNDERLINE)

New DragMode (GPDRAGMODE) property controls grid behavior when dragging within a grid: 0=change active cell; 1=use drag&drop; 2=extend selection; 3=disable dragging.

drag & drop (source and/or target cell position is reported in GEDRAGDROP event)

Tab

each tab may have an icon (GPICON)

FontUnderline (GPFONTUNDERLINE)

drag & drop (source and/or target tab index is reported in GEDRAGDROP event)

GPDEFVAL is defined to reset the current tab on form Reset.

Progress Bar

custom foreground & background colors are now supported

new "marquee" style displays continuous moving bar for indeterminate progress
drag & drop (source and/or target progress bar percent is reported in GEDRAGDROP event)

Tree

new "checked" style for multi-select tree

new "hierarchical checked" style for multi-select tree where checking or un-checking a node propagates the node state to all descendent nodes. Changing the state of a descendent node may cause ancestor nodes to indicate a mixed state (grayed check).

FontUnderline (GPFONTUNDERLINE)

drag & drop (source and/or target node path is reported in GEDRAGDROP event)

Other Enhancements

Drag & Drop - most controls now support DragID and DropIDs properties. Any control with a DragID can initiate a drag-and-drop operation. As the control is dragged over other controls, the target control's DropIDs is searched for an entry that matches the source control's DragID. If a match is found, the cursor is changed to indicate that dropping is permitted. Otherwise the cursor indicates that dropping is prohibited. When the source is dropped over a matching target control, the GEDRAGDROP event is fired. The event is fired on the target control, and the event argument can be queried for the ID of the source control. Other information may be passed in the event arguments, such as the source and/or target tree node, grid cell, character position (edit & combo box controls), list index, etc.

ATGUISORT - sorts items in grid, list box or combo box

ATGUICOPY - same as *Copy built-in menu ID

ATGUICUT - same as *Cut built-in menu ID

ATGUIPASTE - same as *Paste built-in menu ID

Tab Stop property now available in the designer

Pressing the "Application" key on the keyboard will cause a Context (GECONTEXT) event on most controls

wED Editor

The wED editor now has the same tabbed interface style as the main AccuTerm program. Each open document is referenced by a tab. The drop-down arrow at the right of the tab bar can be used to select any tab.

User-defined keywords have enhanced block start/end/mid definitions, so some of the newer BASIC syntax should be easier to recognize.

The Comment/Uncomment commands now respect the "indent comments" preferences set in the editor options.

AccuTerm 7 Update History

Release 7.0d (06-20-2011)

The following bugs have been fixed (all programs):

1) An error was discovered in a component DLL which is used in nearly every part of AccuTerm. The error prevented some kinds of cleanup from occurring when closing a program. No specific bugs have been attributed to this error, however, for correctness, it has now been fixed.

The following bugs have been fixed (main program):

1) When installed using "portable" or "standalone" mode, the WED Editor and GUI Designer would hang when launched. This was due to a missing type library which has now been added.

2) When installed using "portable" or "standalone" mode, sometime the temporary registry entries were not cleaned up when the program closed.

3) If the path for a non-existent directory was entered as the "home directory" during install, at run-time, the home directory would be set to the program directory. Now, if the directory does not exist, when AccuTerm is first run, it will attempt to create the directory and if successful, use the specified directory.

4) The "new connection" wizard now saves the telnet port and SSH port separately. Previously a single port number was saved between executions of the wizard, and the previous value was used as the default the next time the wizard was executed. If the next execution created a session using a different protocol from the previous execution, the wrong port would be shown as the default.

5) The Upload and Download dialogs now save the file transfer protocol selected and use this as the initial selection next time the dialog is opened.

The following scripting bugs have been fixed:

1) The default value for the Right & Bottom arguments of the Session.Select and Session.SetSelection functions has been changed from -1 to -32768. This was done so that these functions could be used to select text in the screen history, which is referenced using a negative value for the row number. Previously, any negative number was treated as "last column" or "last row".

2) The Item property of the FKey object was assigned as the object's "default" property. This fixed an incompatibility with the GET.PROGRAMMED.KEYS subroutine.

3) The Settings.Dialog() function was not properly displaying nodes in the tree for specific panels when the Categories argument was used to limit which panels should be available in the settings tree.

The following GUI bugs have been fixed:

1) The Context event for Option controls was not firing. This has been fixed.

2) The Required property for Option, List (single-select & drop-down) controls was being ignored. This has been fixed.

3) When using MDI applications, selecting a child form by clicking on the MDI tab would corrupt the MDI child window z-order. This has been corrected.

4) A bug in the GUI typeahead function allowed mouse & keyboard events saved in the typeahead buffer to be processed just before a system dialog was displayed (message box, printer setup, font selection, etc.). This could cause a variety of odd behaviors such as the wrong form being activated. Now the typeahead is cleared before showing a system dialog, just like calling ATGUIACTIVATE clears the typeahead.

5) Fixed problem with Option controls which allowed the cursor keys to change the selection when the ReadOnly property is set to True.

The following wED Editor bugs have been fixed:

1) Sometimes, after doing a Format operation, the Enter key would not function until the cursor keys were used to move the current location.

2) When a new item is created using the WED verb from TCL, the Compile tool was disabled until a File->Save was performed. Now the Compile tool is enabled and you will be prompted to save before compiling.

Release 7.0c (03-02-2011)

The following new features have been added (main program):

1) The installer recognizes a new "AUTOUPGRADE" item in the SETUP.INI file to control automatic migration of preferences and files from an earlier version of AccuTerm. When this item is specified, the

"AccuTerm 7 Upgrade" dialog is suppressed when AccuTerm 7 is first used and an earlier version of AccuTerm is detected. See the help file, AccuTerm 7 Reference chapter, for more details.

2) A new subroutine, FTEXPORTDA, has been added to the FTBP file transfer library. This subroutine works similarly to FTEXPORT, except that instead of the source data being read from a file, it is passed as an argument to the subroutine as a standard dynamic array.

3) The wED editor now has an option to show Min/Max/Close buttons on the document window title bar. By default these buttons are not shown as they sometimes do not play nicely with the new tabbed interface.

4) The Close button for the Session Tabs (and Document Tabs in wED) has been moved from the right edge of the bar to the right edge of the tab for which the close action applies.

5) The command-line parser for the WED, GED and FTWIZ verbs has been enhanced to perform simple upper-case conversion for those platforms that normally have this feature (QM & mvBase). On these platforms, if you enter a command entirely in lower-case, and the file exists in the MD / VOC only as upper-case, the entire command is converted to upper-case before processing.

The following bugs have been fixed (main program):

1)Fixed menu Help->Documentation so that it now opens the selected document (it used to simply do nothing).

2) The "AccuTerm 7 Upgrade" process for converting customized menus from previous versions of AccuTerm now works more reliably and accurately.

3) File transfers to Excel (FTD, FTEXPORT, File Transfer Wizard) where a column contained both string and numeric (or string and date) data, the column was "auto-sized" based only on the string data. If the column contained only short strings, it might not be sized correctly to display the numeric or date values.

4) Fixed Page Setup so that the printer orientation works. Also fixed problem causing paper bin and paper size drop-down lists to be empty.

5) Fixed session tab context menu so that if the main menu File->Close item is disabled, the Close item in the context menu is also disabled.

6) In the WED Editor, documents are always saved using standard Windows line ending of CR/LF. Previously, if a document using only LF or only CR for newlines was saved to the host, the CR or LF control characters were stored in the host record instead of converting to AM. This was because the in

transmission to the host, CR/LF were converted to AM, but not CR or LF by itself.

The following new features have been added to the GUI:

1) New toolbar styles allow a toolbar button to display both an icon and caption. The new styles are 2 (small icon with caption) and 3 (large icon with caption).

The following GUI bugs have been fixed:

1) When creating controls at runtime (on the fly) on a visible form, the controls were initially created using a pre-defined location and size and then moved. The problem is that controls initially are visible, and this caused a noticeable flicker. This has been fixed.
2) When using icons and pictures from a web server, if a URL was used more than once, all but the first time, the resulting image was blank. This has been fixed.

3) Fixed a timing issue when images from a web server are being used for grid items or list items. Images from a URL are downloaded asynchronously, and if target row of a icon in a grid or list were to move (insert or delete rows before the target row) before the image was available from the server, it could get stored in the wrong item. This has been fixed.

4) In the HTML Viewer control, setting the Content property more than once failed on all but the first time, and the Reset command was ignored. This has been fixed.

5) Combo box and single-line edit controls allowed multi-line data to be pasted from the clipboard. Now, only the first line of multi-line data is pasted.

6) When copying data to the clipboard from a grid cell with selected text, the entire cell content was copied instead on only the current selection. Now only the selection is copied.

7) When pasting into a grid cell, sometimes the grid status was not marked as "changed".

8) When using MDI applications, if a control on an MDI child form as focus, and a non-MDI form (dialog form) is shown / activated, when the original MDI child form is re-activated, the focus would advance to the next control on the form. This has been fixed.

9) Fixed a refresh problem with dropdown lists using border style of Flat or None.

10) Fixed a problem with the MDI tab bar where a tab would not appear when a new child window is first created. Switching to a different child window and back to the problem window caused the tab to appear.

11) Fixed a small window in the typeahead logic where an event would be queued for transmission to the host, but typeahead would not capture keyboard and mouse events until the transmission was initiated. With very quick fingers it was sometimes possible for keyboard and/or mouse actions to alter the state of a control. This new state would not be reflected in the event already queued, so the host program might get out of sync with the actual runtime state.

12) When the height of a Tab control was very small, an "Invalid procedure call" error was returned. This has been fixed.

13) Setting the Foreground color of a Frame control at run time sometimes ignored the change until the form was refreshed. This has been fixed.

14). Disabling a frame during a List box Activate event may cause the Click event for the list to be lost. This has been fixed.

Release 7.0b (08-26-2010)

The following new features have been added (main program):

1)When the function bar is displayed, the bank selection (leftmost button in the function key button bar) now includes a new choice: *Auto*. When *Auto* is selected, the function key bar tracks the state of the Shift and Ctrl keys. Other selections (Normal, Shift, Ctrl, Shift+Ctrl) lock the bar to the designated bank.

2) The "reconnect" panel that is displayed when a connection is lost has a new "Info" button. Clicking this button will display additional information related to the cause of the disconnect (a popup help topic). Similarly, the connection error dialog that pops up when a network or communication error occurs also includes an "Info" button to display additional information about the error.

3) The character set selected in the Settings dialog is now used as the "User Preferred Supplemental Set" (UPSS) when running the VT220, 320 & 420 emulations. Similarly it determines the international character set used in Prism P30 emulation.

4) The wED editor now has a Maximize menu item in the Window menu to maximize the current document.

5) The Session object now includes AnsiAutoPrint and CharacterMapping properties. These were previously available only as properties of the Settings object.

The following bugs have been fixed (main program):

1) On systems using multiple monitors, any tree control (settings dialog, GUI designer, GUI program using a tree, etc.) placed on a monitor which is to the left of the primary monitor (the one with the Start menu) would not respond to mouse clicks.

2) When using Viewpoint Enhanced emulation with a screen background picture, the picture was not visible when using the ADDS 4000 style "normal" visual attribute. The picture should be visible wherever the "normal" attribute is displayed.

3) In certain cases where the screen is frequently switched between normal (80 column) and extended (132 column) mode, occasionally, when in extended mode, the screen would display blanks in columns past the end of the normal screen width. When the problem occurs, it would usually only affect one or two lines on the screen.

4) The PrinterName and TitleFormat properties caused an error when used in a recorded macro.

5) A small bug which caused the Host Type field in the Settings dialog to show "custom" instead of the selected host type has been fixed.

6) In the Settings dialog, the list of print adapters may not be properly loaded when starting AccuTerm 7 from a shortcut that does not use Atwin70 as the startup directory. Also, sometimes the adapter selected would not match the one that is used. Finally, there should have been a tooltip that displays the print adapter description for each item in the list when the mouse hovers over any item. These problems have been fixed.

7) After changing certain settings to zero (for example, Interline Delay), then saving the session, the changed setting would not be saved in the session file.

8) In the main AccuTerm program, using the menu designer, it was not possible to disable the main menu.

9) A bug that caused AccuTerm to crash when using [ActiveSession.Close] from a function key was fixed.

10) When opening multiple sessions, especially from a layout file, AccuTerm could get into an infinite loop activating and deactivating the newly added session windows.

11) The script used to install the host programs from the MultiValue -> Host Programs menu failed when AccuTerm was launched from a shortcut that did not use Atwin70 as the startup directory.

12) The FTSETUP program used by the MultiValue -> Host Programs menu (and from TCL prompt) did not update existing catalog items in the MD / VOC when you choose "replace all" from the "item exists"

prompt.

13) When importing a large number of keywords into wED, the import would fail.

14) In the WED Editor, double-clicking a label on a GOTO or GOSUB statement moves the cursor to the line containing the target label. The document should scroll so that the target is in the center of the window. Also, some of the WED dialog boxes (Goto, Find, etc.) would not position themselves on the same monitor as the main window if the monitor containing the main window was to the LEFT of the Start menu (primary monitor). These have been fixed.

15) In the WED Editor, if you have multiple documents open and they are not "maximized", opening a new document switches all documents to "maximized". This has been fixed so that a new document will open in the same window state (maximized or normal) as the currently active document.

16) In the WED Editor, the code formatter was unable to handle using the LOOP keyword as a variable name. This has been partially fixed. However the code formatter is unable to distinguish between LOOP as a keyword and variable name if it is the first word in a line.

17) When the Settings dialog is open, pressing the F1 key usually displayed the Contents page from the help file instead of the help page for the selected category.

18) When using the keyboard with the connection wizard, pressing Enter to each prompt, the newly created session would immediately abort the connection process. This was caused by the Enter key being handled first by the Finish button in the wizard, then by the Abort button on the session window. This has been fixed by delaying the appearance of the Abort button by a second or two.

19) A custom title did not always work - sometimes the title reverted back to "AccuTerm 7".

20) The "Standalone" installation option would fail unless the user had administrator privileges. This has been fixed (one of the main reasons for this installation option is to enable non-admin users to install the product for their personal use).

The following new features have been added to the GUI:

1) A minor enhancement / change to the Resize event: when a form is moved by the user, a Resize event is now fired, even if the size has not changed. The new position is indicated in GUIARGS<1,4> (left) and GUIARGS<1,5> (top). The Resize event is now available to fixed size forms.

The following GUI bugs have been fixed:

1) Several GDI object leaks have been fixed.

2) Under Windows XP only, in the GUI, clicking most controls that have a tooltip would cause the tooltip to never display again. Similarly, leaving the mouse over a control with a tooltip until the tip disappeared would never display the tip again.

3) The way that SelStart and SelLength in GUI combo boxes has been changed. Setting the SelStart property implicitly sets the SelLength property to zero, removing any selection highlight from the text part of the combo box. This is the same behavior as the edit control. A bug in the original release of AccuTerm 7 caused SelLength to be set to a negative value, causing text from the beginning up to the new SelStart position to be selected (if you were to retrieve the SelStart and SelLength properties after setting SelStart, SelStart would be zero and SelLength would be the value you had specified for SelStart). A bug in the 2K2 version of the combo box caused SelLength to be set to the end of the text selecting the text from the SelStart position to the end.

4) In the GUI, the status bar was firing Click, DblClick and Context events whether or not these events were selected in the designer.

5) In the GUI, when a grid was placed in a container whose Enabled property was False, changing the container's Enabled property to True did not cause the grid to become enabled.

6) When an MDI Child form is first displayed, the first control in the form was not automatically activated.

7) In the GUI designer, the Preview function will now work with an MDI app object.

8) In the GUI, when word-wrapping in a grid control is enabled, text in the last column may not be wrapped correctly. This is partially fixed. The underlying problem is that the actual size of the last column changes as the grid is scrolled horizontally. This is not solvable. However, when the grid displays all the columns, the wrapping in the last column should now be correct.

9) Using a Picture control as a container failed for many control types. This was due to the fact that the Picture control does not have BackColor, ForeColor or Font properties, and several control types inherit these properties from their parent. When the picture control is the parent, these properties will be inherited from the Form instead.

10) The GUI runtime maintains a count of visible forms. When any forms are visible, the "this session is running a GUI application" message is displayed on the text screen. There were certain sequences of functions that corrupted this count. This could cause the text screen to be activated when hiding a GUI form while other forms are still visible. This causes a flickering as the text and GUI applications are switched, and degrades performance.

11) When using MDI forms, each child form can have its own main menu. When the child form is the active form, its main menu replaces the main menu of the MDI parent window. A problem was discovered when a new MDI form was being created, before its menu had been initialized, the empty child menu replaced the parent menu. Once the child form's menu was initialized, the new child menu replaced the empty menu. This caused the child forms to quickly "jump" up and down as new forms were created. This is now fixed.

12) When an MDI application has one or more hidden MDI child forms, the tab for the hidden child forms is still displayed on the MDI Tab bar.

13) When using the "multiple application model" with MDI subroutines, after all MDI apps had been deleted, then after showing a modal dialog, the "host busy" window would appear and there was no way to hide it.

14) In the GUI it was possible for the Enabled state of a container control (frame, picture box, tab) to become disabled when it should not.

15) Several bugs related to controls placed directly on a Tab Group control have been fixed. These were all related to focus problems when activating a control using the mouse. Symptoms ranged from missing Validate events, to requiring multiple clicks on command buttons to trigger a Click event. Note: the controls must be in the Tab Group; all controls on a Tab behaved correctly.

16) When deleting the only Tab in a Tab Group, any controls that are placed directly on the Tab Group become hidden when any Tabs are added to the Tab Group. They actually are not hidden, just obscured by the newly added Tabs.

17) When using a non-default font for a List Box control sometimes the vertical scroll bar in the list box was incorrect or missing.

18) Several problems in the GUI designer where certain properties were not saved correctly have been fixed.

19) When positioning a form using "auto" coordinates, which should center the form in the display area, AccuTerm used the actual screen size rather than the usable screen size when calculating the position. This could cause the bottom edge of large forms to be obscured by the Windows task bar.

20) Under Windows Vista and Windows 7, the frame of fixed-size forms is the same as for sizable forms. This is different behavior than Windows XP. To make this difference "transparent" , Windows reports the size smaller than it actually is (Windows Vista & 7 report lie about the border size and return the same value that was returned in XP). This could cause forms positioned at or near zero for the left or top coordinate to have their border displayed off-screen, or on the adjacent monitor if using multiple monitors. AccuTerm now adjusts for this different behavior.

21) Printing a form that contained an HTML Viewer control did not print the Viewer control. This has been fixed, however, due to the way the Viewer is implemented, printing quality may poor and pixilated.