SOFTWARE DEVELOPERS MANUAL - Properties and Methods for SigPlus®

Version 3.74 - 9/11/06 -- FOR TECHNICAL SUPPORT CALL 805 520-8286
OR E-MAIL TO: topaz_support@topazsystems.com **PN:09000008-10**

3.73 Optimized SigPlus timing for increased performance.

3.69 Optimized HID USB communication.
Optimized serial communications when used in thin client environment
Added pressure data functionality (hardware-specific functionality)
Added ImageScreenResolution() for setting resolution of images written by SigPlus
Added support for SE pads (SE pads require SigPlusSE install)

3.64 Added CompressionMode 100, for maximizing signature compression
Added support for new Adobe signer, GemSignPlus. Do rotation much differently

3.61 Disabled unused SigPlus OnClick event to improve application performance.

3.60 Added GetBitmapBufferBytes() method to return a Bitmap image in a Byte array. Added tablet connect query for serial and HSB only This checks whether or not the tablet is connected to the computer. Added SetEnableColor() method to activate color in SigPlus..

3.55 Added DisplayRefreshInterval() method to allow developer to specify the SigPlus refresh rate. Added SetBackgroundHandle () method to allow SigPlus background to be set from a windows handle.

3.53 Added functionality to allow the font used for annotation within SigPlus to be specified (section must be added in the SigPlus.ini). Further improved HID tablet performance.

3.49 Eliminated issues when trying to import signatures into an instance of SigPlus in which SigCompressionMode does not match

3.48 Improved HSB (HID) tablet performance.

3.46 Implemented GetSigPlusVersionString() method for determining version of SigPlus currently installed. Implemented GetLCDSize() and GetLCDTextSize(str String) to help determine LCD area necessary for given string size to be displayed on Topaz LCD tablets. Added four methods to create buffered signature BMPs (method calls all begin with "BitmapBuffer"), eliminating the need to save them to a file location. Updated ClearTablet method to increase efficiency when called within Pen Events.

3.40 Eliminates "flicker" when writing on older operating systems. Added the DisableMessageBoxes method, to inhibit SigPlus error messaging.

3.34 Implemented LCDWriteFile method, allowing BMP to be passed to LCD tablet using its full path (rather than a Handle to the BMP)

3.33 Internal changes. No major modifications.

3.32 Added SigPlusInit() method for use with CreateObject method in vbscript on a web page.

3.29 Implemented PenUp and PenDown events (see SetEventEnableMask method). Implemented ability to query tablet for serial number and model number (see TabletSerialNumber and TabletModelNumber methods). For use with specific hardware only.

3.28 Provides interactive LCD tablet functionality when minimized by updating the signature list when KeypadQueryHotSpot, NumberOfTabletPoints, and TabletState methods are called.
Input buffer is cleared on open, safeguarding against leftover data issues.

3.23 Timeout changed in SigPlus to accommodate Citrix system performance, allowing compatibility with Citrix environment. Added SetBackground method, allowing BMP image to be set as background of SigPlus object
Added parameter to SigPlus.ini, to permit multiple USB tablets on a hub to sign simultaneously. Added parameter to SigPlus.ini to allow SigPlus error messages to be suppressed. Fixed bug to allow SigPlus to be unregistered using regsvr32.

For information on versions of SigPlus older than 3.23, please contact Topaz Systems, Inc.
For updates to this manual and to software, check www.topazsystems.com/SoftwareFor examples and answers to frequently asked questions, see www.topazsystems.com/Software/faq.htm

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

GENERAL RELEASE NOTES

Setting the COM Port:

Be sure that tablet state is off when selecting com port. Com port must be selected first, and then tablet state turned on. Also be sure to set tablet state off before exiting the application. **The Com port can also be set in the SigPlus.ini file which is configured on installation of the OCX.**

BSTR Data Transfer:

To transfer data using BSTR methods, you may wish to refer to the Microsoft KB article ID # Q103257, ACC: Reading, Storing, and Writing Binary Large Objects (BLOBS).

The preferred method to transfer signature data into and out of a database, over a network, or the Internet is to use the SigString property. This represents the signature in characters that only consist of the ASCII numbers 0-9 and capital letters A-F.

Very Important Note for non-C++ Users:

For all Properties, the function names are in the form of a Get/Set pair. The full name is used when the object is referenced from a Visual C++ application. For Visual Basic, Powerbuilder, and other users, just use the property name itself, **without the Get/Set prefix (NOTE: Method calls in VB still require the Get prefix…see the specific Remark section for each method for details)** Note that in version 2.30, the add 32 TabletMode was added to provide lower-resolution signatures sometimes needed by VB. There are other syntax differences between the Visual C++ syntax noted herein and VB or VBA syntax, most notably involving the use of the parenthesis in the argument.

Using the SigPlus.ini feature

The SigPlus.exe program installs the OCX control in the WIN\SigPlus directory, together with the SigPlus.ini file in the WIN directory. After you have installed the .ini file, you may edit it or delete it easily. (This .ini is not functional unless it has been placed in the WIN directory.) If the .ini file is not present, then the properties of the control in your application are unaffected and the application runs on its own. When the .ini file is present in the WIN directory, the values present are used to override the properties set in the control proper in the application. The .ini file allows a common application to use different tablet types and interfaces. For example, if one machine uses the signature pad on COM1, another on COM2, a third on the USB port, and yet a fourth on the USB port with a Topaz HSB tablet, the SigPlus.ini files for these four instances would be:

Computer 1 (COM1)	Computer 2 (COM2)	Computer 3 (USB)	Computer 4 (HSB)
[Tablet]	[Tablet]	[Tablet]	[Tablet]
TabletComPort=1	TabletComPort=2	TabletComPort=(*)	TabletComPort=(*)
TabletType=0	TabletType=0	TabletType=2	TabletType=6
TabletLCDMode=0	TabletLCDMode=0	TabletLCDMode=0	TabletLCDMode=0
etc.	etc.	etc.	etc.

* - TabletComPort setting in these instances does not matter

It is very important to note that the .ini settings can only be used to override the OCX default property settings. If coding or script is used to override a property setting in your application, the .ini settings will not override the scripted setting.

TabletType settings: Default is 0

0. Normal mode. When tablet is activated it will accept input from the selected com port
1 WinTab mode,when the tablet is activated, it will accept data from the Topaz WinTab driver only.
2 USB mode, when the tablet is activated, it will accept data from the Topaz USB driver
6 HSB mode (USB tablet using Windows HID driver, models ending in –HSB)

The TabletComPort and TabletLCDMode configuration lines are identical to the properties with the same names as described in this document. The TabletType configuration line uses that same settings as the lower order settings of the TabletMode property as described later in this document.

Version 3.01 and above of the SigPlus control allows the background and ink color of the control to be set under program or property control. To enable these properties, set the line EnableColor = 1 in the [Tablet] portion of the SigPlus.ini file at the top.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

To use a USB tablet on Windows 95, please set the INI parameter Win95USB=1, and then install the older Topaz USB drivers, Topazdrv.sys and Topaz.inf. For all other Win OS, leave Win95USB=0, and install instead the TopazUsb.sys and TopazUsb.inf files for USB (or reinstall SigPlus.exe with these choices).

To create BMP or TIF image files at tablet resolution, please set the ImageScreenResolution=0.

To connect multiple USB tablets to a USB hub, and function independently, please set the UseMultiUSB=1. For a typical, single USB tablet setup, keep UseMultiUSB=0

To suppress error messages generated by SigPlus, please set DisableMessages=1.

Topaz HSB tablets use the built-in HID USB driver rather than a separate Topaz USB driver. No driver loading is necessary since it is already automatically installed. Requires the SigPlus.ini parameter TabletType=6.

NOTE REGARDING TABLET SETTINGS:

Signatures are stored relative to the LogicalX and LogicalY sizes, so it is important to be consistent in all instances of the control. If you change the LogicalX and LogicalY settings, you will need to convert any .sig files saved with the previous LogicalX and LogicalY settings. (LogicalX and LogicalY determined by Start values minus Stop values for that tablet)

SigPlus.ini file parameters for LCD tablets

LCDType	Type of LCD display
LCDXSize	X Size of LCD display, in pixels
LCDYSize	Y Size of LCD display, in pixels
LCDXStart	X Pos in logical tablet coordinates of LCD
LCDYStart	Y Pos in logical tablet coordinates of LCD
LCDXStop	X Pos in logical tablet coordinates of LCD
LCDYStop	Y Pos in logical tablet coordinates of LCD

Installation System Script:

In order to make your own installation script rather than using the Topaz sigplus.exe install, install the following files where noted and then self-register OCXs and DLLs. If you installed the SigPlus control onto your system using the Topaz setup.exe, then the source location for the reference files is the same as the destination location listed below.

OLE32.dll	install to Windows\System\
OLEAUT32.dll	install to Windows\System\
MFC42.dll	install to Windows\System\
MSVCRT.dll	install to Windows\System\
SigPlus.ocx	install to Windows\SigPlus\
SigPlus.ini	install to Windows directory for .ini override of OCX tablet and Com source.
SigSign.ocx	install to Windows\SigPlus\

DemoOcx.exe (The SigPlus demo program) will require these files in addition to those above:

COMDLG32.dll	install to Windows\System\
COMCTL32.dll	install to Windows\System\
OLEDLG.dll	install to Windows\System\

Wrappers:

For Visual C++ programmers, the SigPlus wrappers are provided in the installation of the software. By default, the SigPlus.cpp and SigPlus.h files are provided in WIN\sigplus\wrappers.

Important Notice:

These guidelines or any or all additional documentation or examples do not constitute a warranty about the performance, security, or legal acceptability of SigPlus software in any specific use or implementation. To the extent that SigPlus is used to achieve regulatory or other specific objectives within an industry, you must consult competent experts or regulatory officials together with your own plan to achieve your desired business objectives using the Topaz tools.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

EVENTS FOR SIGPLUS OCX CONTROL:

SigPlus_PenDown

This event will fire when the pen has made contact with the tablet.

SigPlus_PenUp

This event will fire when the pen has been lifted from the tablet.

EVENTS IN SIGPLUS MUST BE USED IN CONJUNCTION WITH THE
SetEventEnableMask(int EventMask)
METHOD OF SIGPLUS. PLEASE REFER TO THIS METHOD IN THE METHODS SECTION FURTHER IN THIS DOCUMENTATION.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

METHODS AND PROPERTIES FOR USE WITH LCD TABLETS

General Methods:

ClearSigWindow (int Location)

Function: Clears pen data either inside or outside the SigWindow (see SetSigWindow).
Arguments: Integer:
Location Specifies clearing inside or outside the SigWindow:
If Location = 0, then signature data is cleared (inside SigWindow)
If Location = 1, then points are cleared outside the SigWindow

Return Value: Void

LCDSetTabletMap (LCDType, LCDXSize, LCDYSize, LCDXStart, LCDXStop, LCDXStop, LCDYStop)

Function: Used to override the default values for the LCD parameters at run time.
Arguments: Integers:

LCDType	Specifies LCD type and format, 0 for 240x128, 1 for 128x64
LCDXSize	X Size of LCD display, in pixels
LCDYSize	Y Size of LCD display, in pixels
LCDXStart	X Pos in logical tablet coordinates of LCD
LCDYStart	Y Pos in logical tablet coordinates of LCD
LCDXStop	X Pos in logical tablet coordinates of LCD
LCDYStop	Y Pos in logical tablet coordinates of LCD

Return Value: Void

Remarks: This method is set at load time by parameters in the SigPlus.ini file, or can be run in the program to override the .ini file settings. Note that LCDType = 1 is not supported in this release.

SetSigWindow(Coords, XPos, YPos, XSize, YSize)

Function: This function sets a window in the logical tablet space that restricts the operation of some functions to the specified window. The functions behave as follows:
JustifyMode will only operate on points inside of this window.
ExportSigFile and WriteImageFile will only operate on points inside the window.
SigString only operates on points inside of the window.
ClearTablet will only clear in the window.

Arguments: Integers:

Coords	Coordinate system used for this hot spot 0 = Logical tablet coordinates, 1 = LCD Coordinates
XPos	Location in logical tablet coordinates (upper left - 0,0)
YPos	Same
XSize	XSize in logical tablet pixels
YSize	YSize in logical tablet pixels

Return Value: Void

Remarks:
This behavior is enabled by setting the start and stop values to non-zero. The window defaults to (0,0,0,0). The window can be enabled at one spot, re-enabled at another, etc., without disabling in between, and then disabled when the various parts of the tablet data have been separated and stored. To determine the logical values in the control for the installed tablet, see the TabletLogicalXSize and TabletLogicalYSize properties.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

Graphics Support: Graphics Methods:

LCDSetFont (Height As Long, Width As Long, Weight As Long, Italic As Integer, Underline As Integer, PitchAndFamily As Integer, FaceName As String)

Function: Sets the size, type, and properties of font used when calling the LCDWriteString method. The arguments are all defined in the LOGFONT data structure (see CreateFont function of Windows API) in Windows for logical fonts.

Arguments: height: Height of font in pixels. width: Width of font in pixels (If 0, the font mapper uses a default width that matches the height.)

Weight: Font weight as a number between 0 and 900. 0=default, 400=normal, and 700=bold.

Italic: If this value is non-zero, the text is italicized.

underline: If this value is non-zero, the text is underlined.

PitchAndFamily: Specifies the pitch (fixed or variable width) and font family used if the font you request is unavailable. If you specify a font that's likely to be, then this argument can be left as 0.

FaceName: Font's name—for example, "Times New Roman", "Courier New", "Arial"

LCDSetWindow (XStart, YStart, XSize, YSize)

Function: This function sets the tablet so that the LCD display will be showing ink only ina restricted area when data is input with a pen in LCDCaptureMode = 2 and = 3 inking modes. Returning the tablet to default state (such as using LCDCaptureMode = 1) will reset these values.

Arguments: Integers:

XPos	Location in LCD coordinates (upper left - 0,0)
YPos	Same
XSize	XSize in LCD pixels
YSize	YSize in LCD pixels

Return Value: True if checksum received and verified. False if no or incorrect checksum received from tablet.

Remarks: Do not send a command with XSize or YSize = 0

GetLCDSize()

Function: Returns the XSize and YSize (in LCD coords) of the tablet currently specified in the SigPlus.ini.

Return Value: unsigned long - (YSize – upper 16 bits, XSize – lower 16 bits)

GetLCDTextSize(str StringToBeDisplayedOnLCD)

Function: Returns the XSize and YSize (in LCD coords) necessary to display on the LCD the string argument, based on the font parameters (see LCDSetFont() method for details). Use to determine the LCD size necessary for your display Text.

Argument: String – Text to be displayed on the LCD

Return Value: unsigned long - (YSize – upper 16 bits, XSize – lower 16 bits)

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

Remarks: Be sure to call the LCDSetFont() method prior to calling GetLCDTextSize(). LCDTextSize() bases it's return value on the sizes set using LCDSetFont().

LCDRefresh (Mode, XPos, YPos, XSize, YSize)

Function: The tablet is sent a refresh command with 4 possible modes

Mode 0 - Clear: The Display is cleared at the specified location.
Mode 1 - Complement: The Display is complemented at the specified location.
Mode 2 - WriteOpaque: The contents of the background memory in the tablet are transferred to the LCD display, overwriting the contents of the LCD display.
Mode 3 - WriteTransparent: The contents of the background memory in the tablet are transferred to the LCD display, and combined with the contents of the LCD display.

Arguments: Integers:

Mode	0, 1, 2, 3 as defined above
XPos	Location in LCD coordinates (upper left - 0,0)
YPos	Same
XSize	XSize in LCD pixels
YSize	YSize in LCD pixels

Return Value: True if checksum received and verified. False if no or incorrect checksum received from tablet.

Remarks: **NOTE**: that this function only can occur on horizontal 8 LCD-pixel boundaries on the LCD tablet unit.

LCDWriteBitmap(Dest, Mode, XPos, YPos, XSize, YSize, Bitmap)

Function: Used to write windows bitmap data to the LCD Display. The data is written at the location specified by the combination of Dest, XPos, and YPos. The Mode determines how the data is written.

Mode 0 - Clear: The Display is cleared at the specified location.
Mode 1 - Complement: The Display is complemented at the specified location.
Mode 2 - WriteOpaque: The contents of the background memory in the tablet are transferred to the LCD display, overwriting the contents of the LCD display.
Mode 3 - WriteTransparent: The contents of the background memory in the tablet are combined with and transferred to the visible LCD memory

Arguments: Integers:

Dest:	0 = Foreground, 1 = Background memory in tablet
Mode	0, 1, 2, 3 as defined above
XPos	Location in LCD coords to draw at
YPos	Same
XSize	Width in LCD pixels
YSize	Height in LCD pixels
Bitmap	Windows handle (HBITMAP) to a bitmap to be displayed. See Object.Handle property in VB Docs

Return Value: True if checksum received and verified. False if no or incorrect checksum received from tablet.

Remarks:

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

LCDWriteFile(Dest, Mode, XPos, YPos, XSize, YSize, Format, FilePath&Name)

Function: Used to write the image data to the LCD Display. The data is written at the location specified by the combination of Dest, XPos, and YPos. The Mode determines how the data is written.

Mode 0 - Clear: The Display is cleared at the specified location.
Mode 1 - Complement: The Display is complemented at the specified location.
Mode 2 - WriteOpaque: The contents of the background memory in the tablet are transferred to the LCD display, overwriting the contents of the LCD display.
Mode 3 - WriteTransparent: The contents of the background memory in the tablet are combined with and transferred to the visible LCD memory

Arguments:	Integers:	
	Dest:	0 = Foreground, 1 = Background memory in tablet
	Mode	0, 1, 2, 3 as defined above
	XPos	Location in LCD coords to draw at
	YPos	Same
	XSize	Width in LCD pixels
	YSize	Height in LCD pixels
	Format	Image file format, see WriteImageFile
	String:	
	FileName:	Path and name of BMP image file to load.

Return Value

Remarks:

LCDWriteString(Dest, Mode, XPos, YPos, XSize, YSize, Format, HexString)

Function: Used to write the image data to the LCD Display. The data is written at the location specified by the combination of Dest, XPos, and YPos. The Mode determines how the data is written. SEE REMARKS BELOW ON THE FORMAT ARGUMENT

Mode 0 - Clear: The Display is cleared at the specified location.
Mode 1 - Complement: The Display is complemented at the specified location.
Mode 2 - WriteOpaque: The contents of the background memory in the tablet are transferred to the LCD display, overwriting the contents of the LCD display.
Mode 3 - WriteTransparent: The contents of the background memory in the tablet are combined with and transferred to the visible LCD memory

Arguments:	Integers:	
	Dest:	0 = Foreground, 1 = Background memory in tablet
	Mode	0, 1, 2, 3 as defined above
	XPos	Location in LCD coords to draw at
	YPos	Same
	XSize	Width in LCD pixels
	YSize	Height in LCD pixels
	Format	Not currently implemented, pass a 0
	String:	
	HexString:	ASCII hex string value.

Return Value: True if checksum received and verified. False if no checksum or checksum incorrect.

Remarks: Currently, the Format argument of this method is non-functional…please pass a 0 for this argument

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

Graphics Properties:

LCDCaptureMode

Function: This property sets the current LCD Mode for the tablet, the tablet is put into the mode as well.

Mode 0 – No LCD Tablet. No LCD commands are sent to the tablet

Mode 1 - Capture Default. CTRL-D is sent to the tablet, which clears the tablet and sets capture mode to be active with Autoerase in the tablet.
Mode 2 - Capture Ink CTRL-T is sent to the tablet, putting the tablet in persistent ink capture mode where the tablet does not automatically clear the display.
Mode 3 - Capture Ink Inverted: CTRL-I is sent to the tablet, where signature ink is displayed inverted against a suitable dark background set using the Graphic functions. Autoerase in the tablet is disabled.

Remarks:

If TabletState is TRUE, the mode command is sent to the tablet immediately. If Tabletstate = false, then this inking mode command is sent when the tablet state is next set to true. When LCDWrite functions, or LCDRefresh are called, and tablet state is TRUE, the mode will automatically be set after completing the Write or Refresh function.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

Keypad Support

Keypad Methods:

KeyPadAddHotSpot(KeyCode, Coords, XPos, YPos, XSize, YSize)

Function: Defines in software the location of a tablet hotspot in logical tablet coordinates. The coordinates of the HotSpot are defined in logical tablet coordinates with (0,0) being the upper left-most pixel. The ini file parameters are used to map the points to logical coordinates if LCD coordinates are used.

Arguments: Integers:

KeyCode	Integer value defining the HotSpot
Coords	Coordinate system used for this hot spot 0 = Logical tablet coordinates 1 = LCD Coordinates.
XPos	Location (upper left - 0,0)
YPos	Same
XSize	XSize in pixels
YSize	YSize in pixels

Return Value: Void

Remarks: The KeyPadAddHotspot() method will require slight variations in px coord location (arguments 3 and 4, varying for about 1 px to 5 px) from its counterpart LCDWriteBitmap(), LCDWriteFile(), or LCDWriteString() method call. For best results, Topaz recommends the following in terms of adding hot spots:

1. Make the hotspot larger than the image/text representing it…this eliminates “hunting and tapping” on the part of the user
2. Making all hotspots no smaller that 10 px in both the Y and X axis

KeyPadQueryHotSpot(KeyCode)

Function: This method queries the data points currently in the control against the logical tablet coordinates mapped by KeyCode. This method returns a true if the control contains data that is within the definition of the KeyCode area on the tablet.

Arguments: Integer

Return Value: Integer, the number of points within the KeyCode definition.

Remarks:

KeyPadClearHotSpotList()

Function: This method clears the controls internal list of hotspots, created using KeyPadAddHotSpot.

Arguments: None

Return Value: None.

Remarks:

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

METHODS FOR SIGPLUS OCX CONTROL:

AddSerialByte(SerialByte)

Function:

Feeds raw data into control as if it came from the tablet, used for alternate data sources..

Argument:

short SerialByte, A single byte to add to the signature.

Return Value:

none

Remarks:

AutoKeyFinish( )

Function:

Completes the auto key generation function. After this call, the key is ready to be used in saving a bound file, and can be retrieved using GetKey().

Argument:

none

Return Value

void

Remarks: PLEASE SEE SetAutoKeyData property in the SigPlus PROPERTY list, to be used in conjunction with this method.

AutoKeyStart( )

Function:

Initializes the automatic key generation function.

Argument:

none

Return Value:

void

Remarks:

PLEASE SEE SetAutoKeyData property in the SigPlus PROPERTY list, to be used in conjunction with this method.

The automatic key generation function will derive a key from the data fed to it via the AutoKeyData property. AutoKeyStart is called to initialize the operation, then AutoKeyData is called repeatedly to input more data to the key generation function. When all of the data has been added, then AutoKeyFinish must be called to complete key generation. This feature can be used to produce a key that is uniquely derived from the input data, and can be used to verify that the data has not changed.

AutoTimeStamp()

Function:

Sets the Time and Date stamp to the current time and date derived from the clock function of the computer.

Argument:

none

Return Value:

none

Remarks:

none

BitMapBufferWrite()

Function:

Creates a bmp file in memory

Argument:

none

Return Value:

none

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

Remarks:
Four methods are used to execute the creation of a buffered BMP: BitMapBufferWrite(),BitMapBufferSize(),BitMapBufferByte(Idx), BitMapBufferClose()

BitMapBufferSize()
Function:
Returns size of the bmp buffer
Argument:
none
Return Value:
long
Remarks:
Use to determine the size of the byte array necessary

BitMapBufferByte(Idx)

Function:
Returns the buffer byte at Idx
Argument:
none
Return Value:
short
Remarks:
Use to return the byte at particular location

BitMapBufferClose()
Function:
Frees the buffer
Argument:
none
Return Value:
none

ClearTablet()
Function:
Causes the control to clear the current signature, and begin a new one. The display is cleared as well as the signature.
Argument:
none.
Return Value:
none
Remarks:

DisableMessageBoxes(int)
Function:
Disables all error messages returned by SigPlus when an error occurs
Argument:
int value: 1=Disable, 0=Enable

DisplayRefreshInterval (Int Milliseconds)
Function:
Set, in milliseconds, the rate at which SigPlus is refreshed during signature capture. Default = 30 ms
Argument:
Int milliseconds
Return Value:
None
Remarks:
None

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

ExportSigFile(Filename)

Function:
The control will write out a signature file in the Topaz image-free raw tablet data vector file format (.sig extension).
Argument:
FileName is a string, containing the path and filename to write to.
Return Value:
bool TRUE if successful, FALSE if not successful
Remarks:
The full Filename must be provided, including the .sig extension.
New in version 2.23 is the ("") argument feature. This allows the control to export the signature to be saved int the control itself. It can be retrieved into the control window using the ImportSigFile ("") command. If a blank signature is exported into the control using (""), the signature-storage contents of the control are erased.

GetBitMapBufferBytes()

Function:
Returns all the buffer bytes
Argument:
none
Return Value:
Byte Array
Remarks:
Use to return all the bitmap bytes at particular location

GetKey(BSTR* KeyData)

Function:
Returns the current binding key, as a BSTR. The returned BSTR must de-allocated by the caller
Argument:
BSTR* KeyData, Pointer to a BSTR containing the raw key data, the number of key data bytes used depends on the algorithm used. See Encryption Mode for more details..
Return Value:
void
Remarks:
This mode is not active

GetKeyReceipt()

Function:
Returns a 32 bit value that is uniquely derived from the key, it can be used to verify that a document has not been modified if the Auto key feature was used to generate the key.
Argument:
none
Return Value:
long 32 bit binary receipt.
Remarks: IN VISUAL BASIC, MUST BE CALLED WITH "GET" PREFIX

GetKeyReceiptAscii( )

Function:
Returns the key receipt as a 8 character ascii hex string
Argument:
none
Return Value:
BSTR The Ascii string
Remarks:
IN VISUAL BASIC, MUST BE CALLED WITH "GET" PREFIX See GetKeyReceipt()

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

GetNumberOfStrokes()

Function: Returns the total number of strokes in the current signature. Can be used to detect if a signature is present, or not.

Argument:

none

Return Value:

short decimal value of number of strokes in the signature.

Remarks:

IN VISUAL BASIC, MUST BE CALLED WITH “GET” PREFIX

A signature may have as few as one stroke. However, the term "stroke" can be interchanged with the term "segment". Since the Topaz tablet is collecting raw tablet data, it cannot determine and does not assume that the data is a signature. The term "stroke" is used to describe the number of segments in the raw data.

GetNumPointsForStroke(StrokeNumber)

Function:

Returns the total number of points in the specified stroke.

Argument:

short StrokeNumber is the number of the stroke to inquire about. Ranges from 0 to NumberOfStrokes - 1

Return Value:

short decimal value of number of points in the stroke.

Remarks: IN VISUAL BASIC, MUST BE CALLED WITH “GET” PREFIX

GetOposPointArray(BSTR* PointArray)

Function:

Returns the Opos representation of the signature in a BSTR . Upon return PointArray will point to a BSTR containing the data in Opos format.

Argument:

BSTR* PointArray, Pointer to a BSTR containing the signature data.

Return Value:

long The total number of points in the array.

Remarks:

IN VISUAL BASIC, MUST BE CALLED WITH “GET” PREFIX. The BSTR returned must be de-allocated by the caller.

Disabled if encryption mode is >/= to 1.

GetOposTotalPoints( )

Function:

Returns the total number of points in the Opos PointArray representation of the signature.

Argument:

none

Return Value:

long The number of points in the array.

Remarks:

IN VISUAL BASIC, MUST BE CALLED WITH “GET” PREFIX The number includes the pen lift points at the end of each stroke.

GetPointXValue(short StrokeIdx, short PointIdx)

Function:

Returns the X coordinate value for the specified point. The value is in LogicalTablet Coordinates.

Argument:

short StrokeIdx, the index of the stroke for the point desired.

short PointIdx, the index of the point in the stroke.

Return Value:

short decimal value of the x coordinate for the point.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

Remarks:
IN VISUAL BASIC, MUST BE CALLED WITH “GET” PREFIX. The Stroke and Point Index must be valid, or 0 is returned.
Disabled if encryption mode >/= 2.

GetPointYValue(short StrokeIdx, short PointIdx)
Function:
Returns the Y coordinate value for the specified point. The value is in LogicalTablet Coordinates.
Argument:
short StrokeIdx, the index of the stroke for the point desired.
short PointIdx, the index of the point in the stroke.
Return Value:
short decimal value of the y coordinate for the point.
Remarks: IN VISUAL BASIC, MUST BE CALLED WITH “GET” PREFIX .
The Stroke and Point Index must be valid, or 0 is returned
Disabled if encryption mode >/= 2.

GetSigData(BSTR* SigData)
Function:
Returns the sig file representation of the signature in the form of a BSTR, that is allocated, filled and returned in *SigData.
Argument:
BSTR* SigData, Pointer to a BSTR containing the signature data.
Return Value:
none
Remarks:
IN VISUAL BASIC, MUST BE CALLED WITH “GET” PREFIX
The BSTR returned must be de-allocated by the caller
When TabletMode = 1024 (400H), the data format is Memo Field, ASCII, and unicode compatible.

GetSigPlusVersionString( )
Function:
Returns the installed version of SigPlus.
Argument:
none
Return Value:
String version of SigPlus (ie, “3.46”)
Remarks: IN VISUAL BASIC, MUST BE CALLED WITH “GET” PREFIX. Can only be used with SigPlus 3.46 or above.

GetSigReceipt( )
Function:
Returns a 32 bit receipt similar to the key receipt. The receipt is formed, by using the auto key generation algorithm on the signature file data. The result can be used to verify that the signature has not been modified
Argument:
none
Return Value:
long 32 bit binary receipt.
Remarks: IN VISUAL BASIC, MUST BE CALLED WITH “GET” PREFIX

GetSigReceiptAscii(short StrokeIdx, short PointIdx)
Function:
Same as GetKeyReceiptAscii, but for Sig receipt.
Argument:
none
Return Value:
BSTR The Ascii string

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

Remarks:
IN VISUAL BASIC, MUST BE CALLED WITH “GET” PREFIX. See GetSigReceipt

GetTrackGemField(short Index)
Function:
Returns the field corresponding to Index from the current TrackGem Signature..
Argument:
short Index, the index of the field desired.
Return Value:
BSTR A string containing the field. This string must be de-allocated by the called.
Remarks:
IN VISUAL BASIC, MUST BE CALLED WITH “GET” PREFIX.
New in version 2.14. Note that LCD Tablet mode must be selected.

HighlightPoint()
Function:
Draws a highlighted circle about the specified point in the currently highlighted stroke..
Argument:
StrokeNumber is a short, the number of the point to highlight about. Ranges from 0 to NumPointsForStroke - 1
Return Value:
none
Remarks:

HighlightStroke(StrokeNumber)
Function:
Draws the specified stroke highlighted.
Argument:
StrokeNumber is a short, the number of the stroke to highlight. Ranges from 0 to NumberOfStrokes - 1
Return Value:
none
Remarks:
none

ImportSigFile(Filename)
Function:
The control will Clear the current signature, read in a signature file in the Topaz vector file format, and display it.
Argument:
FileName is a string, containing the path and filename to read from.
Return Value:
bool TRUE if successful, FALSE if not successful
Remarks:
New in version 2.23 is the (“”) argument feature. This allows the control to import the signature into the control window from the control itself if a signature has been stored in the control using the ExportSigFile (“”) command.
Note that importing signature files into the control is limited to the Topaz vector format files. Image files such as bitmap cannot be imported into SigPlus™. The full Filename must be provided including the .sig extension. SIGPLUS OCX V 1.13 AND LATER USES AND CREATES A MODIFIED .SIG FILE COMPARED TO VERSIONS 1.11 AND EARLIER. TO READ-IN PRIOR .SIG FILES, INVERT THE TABLET LOGICAL Y SETTING.

LoadTrackGemSig(LPCTSTR FileName, short Index)
Function:
Loads the TrackGem signature corresponding to Index from the TrackGem file FileName .

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

Argument:
LPCTSTR
FileName, the TrackGem file to extract the signature from.

Short Index, The index of the signature to load from FileName
Return Value:
BOOL Returns TRUE if the signature was loaded, FALSE if not.
Remarks:
See TrackGem documentation for details of the file format
New in version 2.14

NumberOfTabletPoints()
Function:
Returns the total number of points in the current signature. Can be used to detect if a signature is present, or not.
Argument:
none
Return Value:
short decimal value of number of points in the signature.
Remarks:
A signature should consist of at least 200 points to be considered valid, as this represents approximately 1 second of active signature time. If TabletMode Add 16 is set, then this only reports the number of tablet points that are within the Xstart/Stop and Ystart/Stop boundaries.

SetDisplayAnnotateData(X, Y, Size)
Function:
Sets display screen info for Annotate string.
Argument:
short X, Location to display Annotate string at in signature display window. Uses logical tablet coordinates, relative to the control rectangle.
short Y Location to display Annotate string at in signature display window. Uses logical tablet coordinates, relative to the control rectangle.
short Size, size to display Annotate string, in logical tablet coordinate height.
Return Value:
none
Remarks:
The default is the lower right corner, at ~8% of the screen high.
Note that in version 2.25, the Annotation string can have multiple lines. You may wish to position the annotation string at the top of the control rectangle instead of at the bottom. To do this, set DisplayAnnotateData to the same height as the chosen text height. For example, for the default text height of 8% (120 logical tablet coordinates), setting the Y location to 120 will position the annotation string to start in the upper right corner of the control.

SetDisplayTimeStampData(X, Y, Size)
Function:
Sets display screen info for Time and Date stamp.
Argument:
short X, Location to display TimeStamp string at in signature display window. Uses logical tablet coordinates, relative to the control rectangle.
short Y Location to display TimeStamp string at in signature display window. Uses logical tablet coordinates, relative to the control rectangle.
short Size, size to display TimeStamp string, in logical tablet coordinates high.
Return Value:
none
Remarks:
The default is the lower left corner, at ~8% of the screen height..

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

SetEventEnableMask(int EventMask)

Function:
Enables Pen Up and Pen Down events in SigPlus.

Argument:
short 1 = Pen Down
2 = Pen Up

Return Value:
none

Remarks:
IN VISUAL BASIC, USE "SET" PREFIX. Once the pen event has fired, the SetEnableEventMask method must be called again before the event will once again fire.

SetImageAnnotateData(X, Y, Size)

Function:
Sets image file info for Annotate string.

Argument:
short X, Location to draw Annotate string at in signature image file. Coordinates range from 0 to TabletLogicalXSize - 1.
short Y, Location to draw Annotate string at in signature image file. Coordinates range from 0 to TabletLogicalYSize - 1..
short Size, size to draw Annotate string, in pixels high.

Return Value:
none

Remarks:
The default is the lower left corner, at ~8% of the image high..

SetImageTimeStampData(X, Y, Size)

Function:
Sets image file info for Time and Date stamp.

Argument:
short X, Location to draw TimeStamp string at in signature image file. Coordinates range from 0 to ImageXSize - 1.
short Y, Location to draw TimeStamp string at in signature image file. Coordinates range from 0 to ImageYSize - 1..
short Size, size to draw TimeStamp string, in pixels high.

Return Value:
none

Remarks:
The default is the lower left corner, at ~8% of the image high..

SetKey(BSTR* KeyData)

Function:
Sets the binding key for storing the signature data

Argument:
BSTR* KeyData, Pointer to a BSTR containing the raw key data, the number of key data bytes used depends on the algorithm used. See Encryption Mode for more details..

Return Value:
void

Remarks:

SetSigData(BSTR* SigData)

Function:
Loads the signature as if it were a sig file, from the data in the BSTR.

Argument:
BSTR* SigData, Pointer to a BSTR containing the signature data.

Return Value:
none

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

Remarks:
When TabletMode = 1024 (400H), the data format is Memo Field, ASCII, and unicode compatible.

SetBackground(Filename, Mode)
Function:
Sets a BMP image as the background to a SigPlus control.
Argument 1:
FileName is a string, containing the BMP path and filename to write.
Argument 2:
Int value, not implemented at this time, so pass a 0
Remarks:
You will need to set TabletState=1 prior to calling this method. In addition, set the DisplayWindowRes=True prior to calling this method in Visual Basic. BMP will display at true size.

SetBackgroundHandle(Windows Image Handle, Mode)
Function:
Sets a BMP image as the background to a SigPlus control.
Argument 1:
Windows Handle to an image, containing the BMP path and filename to write.
Argument 2:
Int value, not implemented at this time, so pass a 0
Remarks:
You will need to set TabletState=1 prior to calling this method. In addition, set the DisplayWindowRes=True prior to calling this method in Visual Basic. BMP will display at true size.

SetEnableColor(int)
Function:
Sets SigPlus up for color Signatures: Background and Foreground Color.
Argument:
0 = off, 1 = on, Default = 0

SigPlusInit( )
Function:
Sets SigPlus up properly for use with CreateObject() method in vbscript on a web page. Call prior to any other methods/properties.
Remarks:
No need to call this method using CreateObject anywhere except on a web page (such as HTM or ASP page).

TabletConnectQuery()
Function:
Determines whether Tablet is connected or not
:Return Value:
bool TRUE if conected, FALSE if not connected

ImageScreenResolution()
Function:
Sets SigPlus up to write images at TabletResolution (settable by setting the TabletResolution property in SigPlus), or at ScreenResolution (the current resolution of the target machine's screen)
NOTE: If TabletResolution is to be used, and you change it, make sure it is set to the default value for actual signature capture, or capture results will vary!
Return Value:
int 0=TabletResolution, 1=ScreenResolution

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

WriteImageFile(Filename)

Function:

The control will write out a signature file in the current Image file format. The default is .BMP.

Argument:

FileName is a string, containing the path and filename to write to.

Return Value:

bool TRUE if successful, FALSE if not successful

Remarks:

See the property “SetImageFileFormat” to set an image type other than BMP. Other supported image formats include JPG, TIF, WMF, EMF. The full Filename must be provided, including the standard file extension.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

PROPERTIES FOR SIGPLUS OCX CONTROL:

General Properties:

SetAsciiDataMode()

Function:

This is the Memo Field mode, where the GetSigData and SetSigData formats are unicode compliant for applications using a memo field to store the signature cannot accept binary data. This Memo Field mode is active for all .sig file formats, including normal, cryptographically bound, and compressed. If you save the .sig data in this mode, you must also retrieve the data in the same mode.

Argument:

BOOL Value for AsciiDataMode
TRUE AsciiDataMode mode = active
FALSE AsciiDataMode mode = inactive

Return Value: :

none

Remarks:

Note: This is the preferred way of setting TabletMode = add 1024

GetAsciiDataMode()

Function:

Gets state of AsciiDataMode.

Argument:

none.

Return Value: :

BOOL Value for AsciiDataMode

Remarks:

Note: This is the preferred way of setting TabletMode = add 1024

SetAnnotate(BSTR String)

Function:

Sets Annotate string for signature.

Argument:

BSTR String to use for Annotate string, an ASCII new line character (LF) will break the text into multiple lines. Not that the annotate string is right justified in the OCX control window. Limited to 512 characters.

Return Value:

none

Remarks:

When enabled, care must be taken as to what is stored in Annotation field. For example, the annotation string should not be used to store a set of measurements relating to biometric information together with a document checksum, hash, or receipt, and the claimed identity of the user. It should also not be used to store a reason for signing or a statement of importance of the signature in relation to a document. One reason to avoid this is that a conflict may occur between the statement and the contents of the document. The annotation can be used to save tablet setup parameters like TabletType, signature compression setting, and the logical X and Y settings. Saving the configuration this way is especially useful when different tablets are used in different applications. During signing, the annotation string can be used to display auxiliary information in the vicinity of the signature on the screen, or can be used to display and store the name of the signature box or signer for reference purposes.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

GetAnnotate()

Function:

Gets Annotate string stamp for signature display.

Argument:

none.

Return Value:

BSTR Annotate string.

Remarks:

When enabled, care must be taken as to what is stored in Annotation field. For example, the annotation string should not be used to store a set of measurements relating to biometric information together with a document checksum, hash, or receipt, and the claimed identity of the user. It should also not be used to store a reason for signing or a statement of importance of the signature in relation to a document. One reason to avoid this is that a conflict may occur between the statement and the contents of the document. The annotation can be used to save tablet setup parameters like TabletType, signature compression setting, and the logical X and Y settings. Saving the configuration this way is especially useful when different tablets are used in different applications. During signing, the annotation string can be used to display auxiliary information in the vicinity of the signature on the screen, or can be used to display and store the name of the signature box or signer for reference purposes.

SetAutoKeyData(BSTR* Buffer)

Function:

Adds data to the auto key generation function. If called with file name (and path) when AutoKeyStart has not been initialized, this command will generate AutoKey data from a file rather than adding data via BSTR.

Argument:

BSTR* Buffer, Pointer to a BSTR containing the data, to be added to the key generation.

Return Value:

void

Remarks:

Used with AutoKeyStart and AutoKeyStop methods, but called as a property.

SetCompressionMode()

Function:

Sets the current compression mode for .sig files

Argument:

short integer value as follows:

0	No compression (default)
1	Lossless compression with compacted data format.
2-8	Compression ratio of signature stored in .sig file “2” = 1KB typ. “4” = 500 byte typ. “8” = 250 byte typ.

Return Value:

none

Remarks:

When loading a .sig file, the compression mode must be set to the same value that was used when the .sig file was created. Do not use compression with MicroGem tablet data collection.

GetCompressionMode()

Function:

Returns the current setting of the compression mode for .sig files.

Argument:

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

none

Return Value:

short integer value of .sign fil compression mode as shown above.

Remarks:

SetEncryptionMode(short Mode)

Function:

Sets Encryption mode. This function is used to set the encryption mode used for importing and exporting sig files.

Argument:

short Mode

0	Clear text mode
1	40-bit DES. If a longer key is set, only 40 bits used
2	Higher security encryption mode

Return Value:

none

Remarks:

If Encryption mode >/= 2, the key can only be set if there are no points in the signature (ClearTablet). When the signature has points while in encryption mode 2, the encryption mode cannot be changed to another mode unless the points are cleared.

GetEncryptionMode()

Function:

Gets the current value for the encryption mode.

Argument:

none.

Return Value:

short Current encryption mode.

Remarks:

See SetEncryptionMode

SetJustifyMode(short Mode)

Function:

Sets Justification mode.

Argument:

short Mode

0	Normal, no justification
1	Justify and zoom signature (upper-left corner)
2	Justify and zoom signature (upper-right corner)
3	Justify and zoom signature (lower-right corner)
4	Justify and zoom signature (lower-left corner)
5	Justify and zoom signature (center of control)
Add 16	Justify based on median value of signature data
Add 32	Justify based on mean value of signature data
Add 48	Justify based on mode value of signature data (These modes can recreate a synthetic baseline To Autojustify the median value of the signature data to the center of the OCX box, you would set JustifyMode to 16 + 5 = 21.)
6	Justify Upper Left Scale
7	Justify Upper Left Scale
8	Justify Upper Left Scale
9	Justify Upper Left Scale
10	Scale without justify of signature.

Return Value:

none

Remarks:

When using JustifyMode 1-5 and Add modes, a border area around the signature limits can be set in logical tablet coordinates using the JustifyX and JustifyY

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

values. In modes 6-9, JustifyX = width of the control in inches x 1000.
If JustifyX = 0 when used with modes 6,7,8,9,10 and Acrobat, automatic 1:1 scaling will be set for the transfer of the signature into Acrobat.

GetJustifyMode()
Function:
Gets Justification Mode
Argument:
none
Return Value:
short Current Justification Mode.
Remarks:
See SetJustifyMode()

SetJustifyX(short XPos)
Function:
Sets Justification point X coordinate.
Argument:
Short XPos X coord in Logical Tablet coordinates
When JustifyMode = 1-5, sets autojustify X border
In modes 6-10, JustifyX = width of the control in inches x 1000
Return Value:
none
Remarks:
Implemented for Autojustify mode only

GetJustifyX()
Function:
Gets Justification point X coordinate
Argument:
none
Return Value:
short X coord in Logical Tablet coordinates.
Remarks:
Implemented for Autojustify mode only

SetJustifyY(short YPos)
Function:
Sets Justification point X coordinate.
Argument:
Short YPos Y coord in Logical Tablet coordinates
When JustifyMode = 1-5, sets autojustify X border
In modes 6-10, JustifyY is note used
Return Value:
none
Remarks:
Implemented for Autojustify mode only

GetJustifyY()
Function:
Gets Justification point Y coordinate
Argument:
none
Return Value: :
short Y coord in Logical Tablet coordinates.
Remarks:
Implemented for Autojustify mode only

GetKeyString(BSTR* SigData)
Function:
Provides from the control in Ascii Data (VB script) compatible format, the binding

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

key.

Argument:
none

Return Value:
BSTR* SigData, Pointer to a BSTR containing the signature data.

Remarks:
Can be used to move keys from one instance of the control to another.

SetOpaqueMode(Mode)

Function:
Controls whether the signature is displayed on an opaque background, or transparently.

Argument:
bool
TRUE Use opaque background
FALSE Use transparent background.
Defaults to TRUE

Return Value:
none.

Remarks:
Transparent background is only supported in Visual C++ applications. In VB and VBA applications windows causes the transparent background of the control to shine through to the desktop.

GetOpaqueMode()

Function:
Gets the value of Opaque Mode

Argument:
none

Return Value:
bool
TRUE Use opaque background
FALSE Use transparent background.

Remarks:

SetSaveSigInfo(SaveSigInfo)

Function:
Enables/Disables the saving of TimeStamp and Annotate data in the sig file

Argument:
BOOL Value for SaveSigInfo
TRUE The SigInfo will be saved (default)
FALSE The SigInfo will not be saved

Return Value:
none

Remarks:
By default, the SaveSigInfo property is disabled. When enabled, care must be taken as to what is stored in Annotation field. For example, the annotation string should not be used to store a set of measurements relating to biometric information together with a document checksum, hash, or receipt, and the claimed identity of the user. It should also not be used to store a reason for signing or a statement of importance of the signature in relation to a document. One reason to avoid this is that a conflict may occur between the statement and the contents of the document. The annotation can be used to save tablet setup parameters like TabletType, signature compression setting, and the logical X and Y settings. Saving the configuration this way is especially useful when different tablets are used in different applications. During signing, the annotation string can be used to display auxiliary information in the vicinity of the signature on the screen, or can be used to display and

store the name of the signature box or signer for reference purposes.

SetSigString(BSTR* SigData)

Function:

Loads sig data into the control in Ascii Data (VB script) compatible format. Data is in the form of an Ascii BSTR.

Argument:

BSTR* SigData, Pointer to a BSTR containing the signature data.

Return Value:

none

Remarks:

Used to retrieve or place sig data in the control as a property rather than as a method. The data format is Memo Field, ASCII, and unicode compatible.

GetSigString(BSTR* SigData)

Function:

Gets sig data from the control in Ascii Data (VB script) compatible format. Data is in the form of an Ascii BSTR.

Argument:

none

Return Value:

BSTR* SigData, Pointer to a BSTR containing the signature data.

Remarks:

Used to retrieve or place sig data in the control as a property rather than as a method. The data format is Memo Field, ASCII, and unicode compatible.

SetTimeStamp(TimeStamp)

Function:

Sets Time and Date stamp for signature.

Argument:

BSTR String to use for TimeStamp string, an ASCII new line character (LF) will break the text into multiple lines. Not that the TimeStamp string is left justified in the OCX control window. Limited to 512 characters.

Return Value:

none

Remarks:

GetTimeStamp()

Function:

Gets Time and Date for signature display.

Argument:

none.

Return Value:

BSTR Time and Date string.

Remarks:

SetZoomPower(short Power)

Function:

Sets Zoom power.

Argument:

Short Power The magnification power to use, 1 = normal.

Return Value:

none

Remarks:

Not Implemented in this version

GetZoomPower()

Function:

Gets Zoom power

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

Argument:
none
Return Value:
short Current Zoom power.
Remarks:
See SetZoomPower()
Not Implemented in this version.

SetZoomX(short XPos)
Function:
Sets Zoom point X coordinate.
Argument:
Short XPos X coord in Logical Tablet coordinates
Return Value:
none
Remarks:
Not Implemented in this version
GetZoomX()
Function:
Gets Zoom point X coordinate
Argument:
none
Return Value:
short X coord in Logical Tablet coordinates.
Remarks:
Not Implemented in this version

SetZoomY(short YPos)
Function:
Sets Zoom point X coordinate.
Argument:
Short YPos Y coord in Logical Tablet coordinates
Return Value:
none
Remarks:
Not Implemented in this version

GetZoomY()
Function:
Gets Zoom point Y coordinate
Argument:
none
Return Value: :
short Y coord in Logical Tablet coordinates.
Remarks:
Not Implemented in this version

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

Tablet Properties:

SetTabletBaudRate(Rate)

Function:

Sets Baud rate of tablet.

Argument:

long — 19200 for SignatureGem™ and MicroGem™ tablets
9600 for ClipGem™ tablets.
19200 = default

Return Value:

none.

Remarks:

GetTabletBaudRate()

Function:

Gets Baud rate of tablet.

Argument:

none

Return Value:

long — 19200 or 9600

Remarks:

SetTabletClipping()

Function:

Sets mode where signature points are not reported if the points are drawn outside the XStart/Stop, and YStart/Stop window. When active, pen down writing ourside the Start/Stop window will return zero points as the number of tablet points.

Argument:

BOOL — Value for TabletClipping
TRUE — TabletClipping mode = active
FALSE — TabletClipping mode = inactive

Return Value: :

none

Remarks:

Note: This is the preferred way of setting TabletMode = add 16

GetTabletClipping()

Function:

Gets state of TabletClipping.

Argument:

none.

Return Value: :

BOOL — Value for TabletClipping

Remarks:

Note: This is the preferred way of setting TabletMode = add 16

SetTabletComPort(PortNumber)

Function:

Sets the COM port to use.

Argument:

short — PortNumber is an integer value from 1 to 99.

Return Value:

none

Remarks:

The SigPlus™ OCX control does not lock up a port as is the case with mouse-type drivers. The port is only used when tablet is selected on. Take care to only set COM port when tablet state is OFF.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

GetTabletComPort()

Function:

Gets the COM port in use.

Argument:

none.

Return Value:

short PortNumber is an integer value.

Remarks:

SetTabletComTest()

Function:

Sets Hardware check mode. When this mode is active, Normal mode is also set. If Topaz tablet is plugged into selected COM port, TabletState can be set to 1. If tablet is not plugged into serial port, TabletState cannot be set to 1

Argument:

BOOL Value for TabletComTest
TRUE TabletComTest mode = active
FALSE TabletComTest mode = inactive

Return Value: :

none

Remarks:

Note: This is the preferred way of setting TabletMode = add 128

GetTabletComTest()

Function:

Gets state of TabletComTest.

Argument:

none.

Return Value: :

BOOL Value for TabletComTest

Remarks:

Note: This is the preferred way of setting TabletMode = add 128

SetTabletFilterPoints(Count)

Function:

Sets the number of samples to Filter in order to optimize the smoothness of the captured signature.

Argument:

short integer value of filter points. Default = 4

Return Value:

none

Remarks:

This property is nomally set to 4 (default) for the SignatureGem™ products and is set to 2 for the MicroGem™ and ClipGem™ products.

GetTabletFilterPoints()

Function:

Gets the number of samples to Filter that is set.

Argument:

none

Return Value:

short integer value of filter points.

Remarks:

See SetTabletFilterPoints above.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

SetTabletInvisible()

Function:

Sets mode where control becomes invisible and does not draw visibly.

Argument:

BOOL Value for TabletInvisible
TRUE TabletInvisible mode = active
FALSE TabletInvisible mode = inactive

Return Value: :

none

Remarks:

Note: This is the preferred way of setting TabletMode = add 64

GetTabletInvisible()

Function:

Gets the state of TabletInvisible.

Argument:

none.

Return Value: :

BOOL Value for TabletInvisible

Remarks:

Note: This is the preferred way of setting TabletMode = add 64

SetTabletLCDMode(Mode)

Function:

Sets to correct communication protocol for MicroGem[tm] and MicroGem[tm]*LCD* tablets.

Argument:

bool
TRUE indicates tablet is Topaz MicroGem[tm] series (LCD option).
FALSE indicates other Topaz tablet.
Defaults to FALSE

Return Value:

none.

Remarks:

GetTabletLCDMode()

Function:

Gets LCD Tablet Mode

Argument:

none
Return Value:
bool
TRUE indicates tablet is Topaz MicroGem[tm] series (LCD option).
FALSE indicates other Topaz tablet.

Remarks:

See SetTabletLCDMode property above.

SetTabletLogicalXSize(Xsize)

Function:

Sets the Range of horizontal values to be used in representing signatures. This has no relation to the displayed, image file, or tablet sizes.

Argument:

short integer value of Tablet logical size. Default is 2150.

Return Value:

none.

Remarks:

This is the X-range used for the Topaz vector format, and the internally used format. In most applications, this should be set to match the active tablet X resolution.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

GetTabletLogicalXSize()

Function:

Returns the Tablet Logical X Size.

Argument:

none.

Return Value:

short integer value of tablet logical X size.

Remarks:

SetTabletLogicalYSize(Ysize)

Function:

Sets the Range of vertical values to be used in representing signatures. This has no relation to the displayed, image file, or tablet sizes.

Argument:

short integer value of Tablet logical size. Default is 1400.

Return Value:

none

Remarks:

This is the Y-range used for the Topaz vector format, and the internally used format. In most applications, this should be set to match the active tablet Y resolution.

GetTabletLogicalYSize()

Function:

Returns the Tablet Logical Y Size.

Argument:

none.

Return Value:

short integer value of tablet logical Y size.

Remarks:

SetTabletMode(Mode)

Function:

Determines if the tablet will accept data from a com port, or from the WinTab driver if WinTab support is desired.

Argument:

short Mode

0. Normal mode. When tablet is activated it will accept input from the selected com port
Default is 0.

1 WinTab mode,when the tablet is activated, it will accept data from the Topaz WinTab driver only.

2 USB mode, when the tablet is activated, it will accept data from the Topaz USB driver

3 TracGemPOST signature format, Transparent Mode (CTRL T)

4 Older-model SigLite touch tablet format…obsolete mode

Preferred method of setting the modes above is with TabletType property.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

For Reference only:

Add 32	Rendering of signatures is lower resolution for better Visual Basic compatibility. **Preferred method of setting this mode is with DisplayWindowRes property.**
Add 64	Control becomes invisible and does not draw visibly (0X40) **Preferred method of setting this mode is with TabletInvisible property.**
Add 128	Hardware check mode. When this mode is active, Normal mode is also set. If Topaz tablet is plugged into selected COM port, TabletState can be set to 1. If tablet is not plugged into serial port, TabletState cannot be set to 1. **Preferred method of setting this mode is with TabletComTest property.**
Add 256	Allows for signature rotation in the control after capture. Does not save the .sig file rotated. Display rotation only. **Preferred method of setting this mode is with DisplayRotate property.**
Add 768	Allows for signature rotation in the control after capture. Does save the .sig file rotated in this mode. **Preferred method of setting this mode is with DisplayRotateSave property.**
Add 1024	(400H). This is the Memo Field mode, where the GetSigData and SetSigData formats are unicode compliant for applications where the field use to store the signature cannot accept binary data. This Memo Field mode is active for all .sig file formats, including normal, cryptographically bound, and compressed. If you save the .sig data in this mode, you must also retrieve the data in the same mode. **Preferred method of setting this mode is with AsciiDataMode property.**

Add modes can be used in any additive combination. For example, to implement Add 32 in WinTab mode, the argument is set to 1 + 32 = 33.

Return Value:
none.
Remarks:
If WinTab support is not available, it will not do anything when in the active state. Conversely, if WinTab is present and the mode is not correct, the tablet will also do nothing when active, because the WinTab driver takes exclusive possession of the Com Port

GetTabletMode()
Function:
Returns the current tablet mode of the control.
Argument:
none.
Return Value: short
Remarks:
See SetTabletMode()

SetTabletRotation(short TabletRotation)
Function:
Sets the orientation for display of tablet data. The data in the sig file is stored in the native tablet orientation.
Argument:
short Tablet rotation in degrees, allowed values are:
0
90
180
270
Return Value:
none
Remarks:
Implemented new in version 2.22

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

GetTabletRotation()

Function:

Gets current tablet orientation.

Argument:

none.

Return Value: :

short Current tablet rotation in degrees.

Remarks:

See SetTabletRotation()

SetTabletState(State)

Function:

Set the capture state of the control. When the control is active, pen data is captured and added to the current signature.

Argument:

Short	State
1	Active Attaches the control to the COM port and starts accepting data.
0	Inactive. Detaches from the port and stops gathering data. Default state is 0.

Return Value:

None. Tablet State is retrieved by using GetTabletState. This can be used for tablet connect or driver detect logic. See TabletMode for further information

Remarks:

Only one tablet control can be active on a given serial port at any point in time.

GetTabletState()

Function:

Returns the current capture state of the control.

Argument:

none.

Return Value:

short 1 if accepting data, 0 if not.

Remarks:

SetTabletTimingAdvance(Count)

Function:

Sets the number of timing ticks set into software to match the tablet being used. This is an internal Topaz function.

Argument:

short integer value of timing ticks. Default = 4

Return Value:

none

Remarks:

This property is nomally set to 4 (default) for the SignatureGem™ products and is set to 2 for the MicroGem™ or ClipGem™ products.

GetTabletTimingAdvance()

Function:

Gets the decimal number of timing ticks that are used. (Internal Topaz function).

Argument:

none

Return Value:

short integer value of timing ticks.

Remarks:

See SetTimingAdvance above.

SetTabletType(Mode)

Function:

Determines if the tablet will accept data from a com port, WinTab driver, USB

driver or other method of data input.

Argument:

short Mode

0. Normal mode. When tablet is activated it will accept input from the selected com port. Default is 0.

1 WinTab mode,when the tablet is activated, it will accept data from the Topaz WinTab driver only.

2 USB mode, when the tablet is activated, it will accept data from the Topaz USB driver

3 TracGemPOST signature format, Transparent Mode (CTRL T)

4 Older-model SigLite touch tablet format…obsolete mode

6 -HSB tablet (USB mode for tablets using HID driver)

Return Value:

none.

Remarks:

If WinTab support is not available, it will not do anything when in the active state. Conversely, if WinTab is present and the mode is not correct, the tablet will also do nothing when active, because the WinTab driver takes exclusive possession of the Com Port. This is the preferred way of setting TabletMode for tablet input.

GetTabletType()

Function:

Gets TabletType value.

Argument:

none.

Return Value: :

short integer value of TabletType.

Remarks:

Note: This is the preferred way of setting TabletMode for tablet input.

SetTabletXStart(XStart)

Function:

Sets the X position in tablet coordinates, of the upper left hand corner of the control signature box.

Argument:

short The integer value, in tablet coordinates representing the left-hand edge of the signature box:

See table of recommended values on Page 2 of this document.
Default = 500

Return Value:

none.

Remarks:

To position the starting point of the signature box in tablet space to a particular

spot on the tablet, simply calculate the tablet coordinate of where you wish the left-hand edge of the signature box to start. For example, for the box to be active starting 1 inch in from the left edge on a SignatureGem^{tm}4X5 tablet, set XStart to 500 + 410 = 910.

GetTabletXStart(XStart)

Function:

Gets the X position of the left-hand edge of the active area of the signature box, in tablet coordinates.

Argument:

none

Return Value:

short integer value of tablet X start.

Remarks:

See SetTabletXStart above.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

SetTabletXStop(XStop)

Function:

Sets the X position in tablet coordinates, of the lower right hand corner of the control signature box.

Argument:

short The integer value, in tablet coordinates representing the right-hand edge of the signature box:
See table of recommended values on Page 2 of this document.
Default = 2650

Return Value:

none.

Remarks:

To position the ending point of the signature box in tablet space to a particular spot on the tablet, simply calculate the tablet coordinate of where you wish the right-hand edge of the signature box to stop. For example, for the box to stop being active at 4 inches from the left edge on a SignatureGem^{tm}4X5 tablet, set XStart to 500 + (410x4) = 2140.

GetTabletXStop(XStop)

Function:

Gets the X position of the right-hand edge of the active area of the signature box, in tablet coordinates.

Argument:

none

Return Value:

short integer value ot tablet X stop.

Remarks:

See SetTabletXStart above.

SetTabletYStart(YStart)

Function:

Sets the Y position in tablet coordinates, of the upper-edge of the control signature box.

Argument:

short The integer value, in tablet coordinates representing the upper edge of the signature box:
See table of recommended values on Page 2 of this document.
Default = 350

Return Value:

none.

Remarks:

To position the starting point of the signature box in tablet space to a particular spot on the tablet, simply calculate the tablet coordinate of where you wish the upper left corner of the signature box to start. For example, for the box to be active starting 1 inch below the top usable edge (bottom of the paperguide) on a SignatureGemtm4X5 tablet, set YStart to 575 + 410 = 985.

GetTabletYStart(YStart)

Function:

Gets the Y position of the left-hand edge of the active area of the signature box, in tablet coordinates.

Argument:

none

Return Value:

hort integer value ot tablet Y start.

Remarks:

See SetTabletXStart above.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

SetTabletYStop(YStop)

Function:

Sets the Y position in tablet coordinates, of the lower right hand corner of the control signature box.

Argument:

short The integer value, in tablet coordinates representing the lower edge of the signature box:
See table of recommended values on Page 2 of this document.
Default = 2100

Return Value:

none.

Remarks:

To position the ending point of the signature box in tablet space to a particular spot on the tablet, simply calculate the tablet coordinate of where you wish the right-hand edge of the signature box to stop. For example, for the box to stop being active 3 inches below the top usable edge (bottom of the paperguide on a SignatureGem™4X5 tablet, set XStart to 575 + (410x2) = 1395.

GetTabletYStop(YStop)

Function:

Gets the Y position of the bottom edge of the active area of the signature box, in tablet coordinates.

Argument:

none

Return Value:

short integer value ot tablet Y stop.

Remarks:

See SetTabletXStart above.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

Display Properties:

SetDisplayAnnotate(DisplayAnnotate)

Function:

Enables/Disables the display on the Annotate string in the window

Argument:

BOOL	Value for DisplayAnnotate
TRUE	The Annotate string string will be displayed
FALSE	The Annotate string will not be displayed

Return Value:

none

Remarks:

When enabled, care must be taken as to what is stored in Annotation field. For example, the annotation string should not be used to store a set of measurements relating to biometric information together with a document checksum, hash, or receipt, and the claimed identity of the user. It should also not be used to store a reason for signing or a statement of importance of the signature in relation to a document. One reason to avoid this is that a conflict may occur between the statement and the contents of the document. The annotation can be used to save tablet setup parameters like TabletType, signature compression setting, and the logical X and Y settings. Saving the configuration this way is especially useful when different tablets are used in different applications. During signing, the annotation string can be used to display auxiliary information in the vicinity of the signature on the screen, or can be used to display and store the name of the signature box or signer for reference purposes.

GetDisplayAnnotate()

Function:

Returns the value of DisplayAnnotate.

Argument:

none.

Return Value: :

BOOL	Value of DisplayAnnotate

Remarks:

See SetDisplayAnnotate .

SetDisplayAnnotatePosX(short XPos)

Function:

Sets Display Annotate X location.

Argument:

short XPos	X coord.

Return Value:

none

Remarks:

See SetDisplayAnnotateData()

GetDisplayAnnotatePosX()

Function:

Gets Display Annotate X location.

Argument:

none

Return Value:

short	X coord .

Remarks:

See SetDisplayAnnotateData()

SetDisplayAnnotatePosY(short YPos)

Function:
Sets Display Annotate Y location.
Argument:
short YPos Y coord.
Return Value:
none
Remarks:

See SetDisplayAnnotateData()

GetDisplayAnnotatePosY()

Function:
Gets Display Annotate Y location.
Argument:
none
Return Value:
short Y coord .
Remarks:
See SetDisplayAnnotateData()

SetDisplayAnnotateSize(short Size)

Function:
Sets Display Annotate size.
Argument:
short Size Font size.
Return Value:
none
Remarks:
See SetDisplayAnnotateData()

GetDisplayAnnotateSize()

Function:
Gets Display Annotate Size.
Argument:
none
Return Value:
short Font size.
Remarks:
See SetDisplayAnnotateData()

SetDisplayPenWidth(Width)

Function:
Sets pen width for the displayed signature, in pixels.
Argument:
short integer value for display pen width
Return Value:
None
Remarks:

GetDisplayPenWidth()

Function:
Gets pen width for the displayed signature, in pixels.
Argument:
none.
Return Value: :
short integer value for display pen width
Remarks:

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

SetDisplayRotate()

Function:

Sets mode allowing signature rotation in the control after capture. Does not save the .sig file rotated. Display rotation only. The rotation value is set by TabletRotation.

Argument:

BOOL Value for DisplayRotate
TRUE DisplayRotate mode = active
FALSE DisplayRotate mode = inactive

Return Value:

none

Remarks:

Note: This is the preferred way of setting TabletMode = add 256

Can be used to rotate signatures after capture, if the signature was accidentally taken in a rotation orientation during signature capture. Normally, to change the tablet orientation during capture, the TabletRotation property is used.

GetDisplayRotate()

Function:

Gets state of DisplayPenWidth.

Argument:

none.

Return Value: :

BOOL Value for DisplayRotate

Remarks:

Can be used to rotate signatures after capture, if the signature was accidentally taken in a rotation orientation during signature capture. Normally, so change the tablet orientation during capture, the TabletRotation property is used.

SetDisplayRotateSave()

Function:

Sets mode allowing signature rotation and the save of signature in rotated format after capture. Does not save the .sig file rotated. Display rotation only. The rotation value is set by TabletRotation.

Argument:

BOOL Value for DisplayRotateSave
TRUE DisplayRotateSave mode = active
FALSE DisplayRotateSave mode = inactive

Return Value: :

none

Remarks:

Note: This is the preferred way of setting TabletMode = add 768

Can be used to rotate and then save in rotated format, signatures after capture, if the signature was accidentally taken in a rotated orientation during signature capture. Normally, to change the tablet orientation during capture, only the TabletRotation property is used.

GetDisplayRotateSave()

Function:

Gets state of DisplayRotateSave.

Argument:

none.

Return Value: :

BOOL Value for DisplayRotateSave

Remarks:

Can be used to rotate and then save in rotated format, signatures after capture, if the signature was accidentally taken in a rotated orientation during signature capture. Normally, so change the tablet orientation during capture, only the TabletRotation property is used.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

SetDisplayTimeStamp(DisplayTimeStamp)

Function:

Enables/Disables the display on the Time and Date stamp in the window

Argument:

BOOL	Value for DisplayTimeStamp
TRUE	The TimeStamp string will be displayed
FALSE	The TimeStamp will not be displayed

Return Value:

none

Remarks:

GetDisplayTimeStamp()

Function:

Returns the value of DisplayTimeStamp.

Argument:

none.

Return Value: :

BOOL Value of DisplayTimeStamp

Remarks:

See SetDisplayTimeStamp.

SetDisplayTimeStampPosX(short XPos)

Function:

Sets Display TimeStamp X location.

Argument:

short XPos X coord.

Return Value:

none

Remarks:

See SetDisplayTimeStampData()

GetDisplayTimeStampPosX()

Function:

Gets Display TimeStamp X location.

Argument:

none

Return Value:

short X coord .

Remarks:

See SetDisplayTimeStampData()

SetDisplayTimeStampPosY(short YPos)

Function:

Sets Display TimeStamp Y location.

Argument:

short YPos Y coord.

Return Value:

none

Remarks:

See SetDisplayTimeStampData()

GetDisplayTimeStampPosY()

Function:

Gets Display TimeStamp Y location.

Argument:

none

Return Value:

short Y coord .

Remarks:

See SetDisplayTimeStampData()

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

SetDisplayTimeStampSize(short Size)

Function:
Sets Display TimeStamp size.
Argument:
short Size Font size.
Return Value:
none
Remarks:
See SetDisplayTimeStampData()

GetDisplayTimeStampSize()

Function:
Gets Display TimeStamp Size.
Argument:
none
Return Value:
short Font size.
Remarks:
See SetDisplayTimeStampData()

SetDisplayWindowRes()

Function:
Sets mode which renders signatures in lower (screen) resolution for compatibility in printing directly from VB. MUST BE USED WHEN PRINTING DIRECTLY FROM A VISUAL BASIC FORM.
Argument:
BOOL Value for DisplayWindowRes
TRUE DisplayWindowRes mode = active
FALSE DisplayWindowRes mode = inactive
Return Value: :
none
Remarks:
Note: This is the preferred way of setting TabletMode = add 32

GetDisplayWindowRes()

Function:
Gets state of DisplayWindowRes.
Argument:
none.
Return Value: :
BOOL Value for DisplayWindowRes
Remarks:
Note: This is the preferred way of setting TabletMode = add 32

TabletModelNumber()

Function:
Returns the tablet's internal SigModel number. (Must be used in conjunction with specialized Topaz tablets that support this functionality)
Return Value:
none
Remarks:

TabletSerialNumber()

Function:
Returns the tablet's internal SigSerial number. (Must be used in conjunction with specialized Topaz tablets that support this functionality)
Return Value:
none
Remarks:

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

Image Properties:

SetImageAnnotate(ImageAnnotate)

Function:
Enables/Disables the display on the Annotate string in the image file

Argument:

BOOL	Value for ImageAnnotate
TRUE	The ImageAnnotate string will be drawn
FALSE	The ImageAnnotate will not be drawn

Return Value:
none

Remarks:

GetImageAnnotate()

Function:
Returns the value of ImageAnnotate.

Argument:
none.

Return Value: :
BOOL Value of ImageAnnotate

Remarks:
See SetImageAnnotate .

SetImageAnnotatePosX(short XPos)

Function:
Sets Image Annotate X location.

Argument:
short XPos X coord.

Return Value:
none

Remarks:
See SetImageAnnotateData()

GetImageAnnotatePosX()

Function:
Gets Image Annotate X location.

Argument:
none

Return Value:
short X coord .

Remarks:
See SetImageAnnotateData()

SetImageAnnotatePosY(short YPos)

Function:
Sets Image Annotate Y location.

Argument:
short YPos Y coord.

Return Value:
none

Remarks:
See SetImageAnnotateData()

GetImageAnnotatePosY()

Function:
Gets Image Annotate Y location.

Argument:
none

Return Value:

short Y coord .

Remarks:

See SetImageAnnotateData()

SetImageAnnotateSize(short Size)

Function:

Sets Image Annotate size.

Argument:

short Size Font size.

Return Value:

none

Remarks:

See SetImageAnnotateData()

GetImageAnnotateSize()

Function:

Gets Image Annotate Size.

Argument:

none

Return Value:

short Font size.

Remarks:

See SetImageAnnotateData()

SetImageFileFormat(Format)

Function:

Sets the current format to use for Image files. The default is .BMP. the choices are:

Argument:

short integer value as follows:

0:	Compressed BMP (default)		must have .bmp ext.
1:	Uncompressed BMP		must have .bmp ext.
2:	Mono. BMP		must have .bmp ext.
3:	JPG Q=20		must have .jpg ext.
4:	JPG Q=100		must have .jpg ext.
5:	Uncompressed TIF		must have .tif ext.
6:	Compressed TIF		must have .tif ext.
7:	WMF	(windows metafile)	must have .wmf ext.
8:	EMF	(enhanced metafile)	must have .emf ext.
9:	TIF	(1-bit)	must have .tif ext.
10:	TIF	(1-bit inverted)	must have .tif ext.

Return Value:

none

Remarks:

The file extension is not assumed in the WriteImageFile function. Any extension can be specified, but it should match the specified file format. Note that writing an image file is completely different from a .sig file. An image file is just a standard image format of what is seen in the control and cannot be cryptographically bound or decrypted by the SigPlus control. The .sig file format does not store an image, but rather uses a unique method of preserving the original signature data from the tablet. To create a metafile with a transparent background use a trio of instructions to set TabletOpaque = False, then WriteImageFile, the TabletOpaque = True. For all other image files, TabletOpaque must be true when the image file is written.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

GetImageFileFormat()

Function:

Returns the current setting of the image file format.

Argument:

none

Return Value: short

integer value of image file type as shown above.

Remarks:

SetImagePenWidth(Width)

Function:

Sets the current pen width to use for signature in Image files.

Argument:

short The decimal integer representing the number of pixels of width to make the pen width for image files. Default is 1.

Return Value:

none

Remarks:

This command does not affect the pen width shown in the control signature window. You will notice a natural interaction in perceived pen thickness depending upon the x and y resolution selected for the image.

GetImagePenWidth()

Function:

Gets the current pen width used for signature in Image files.

Argument:

none

Return Value:

short integer value of image pen width

Remarks:

SetImageTimeStamp(ImageTimeStamp)

Function:

Enables/Disables the display on the Time and Date stamp in the image file

Argument:

BOOL Value for ImageTimeStamp

TRUE The TimeStamp string will be drawn

FALSE The TimeStamp will not be drawn

Return Value:

none

Remarks:

GetImageTimeStamp()

Function:

Returns the value of ImageTimeStamp.

Argument:

none.

Return Value: :

BOOL Value of ImageTimeStamp

Remarks:

See SetImageTimeStamp.

SetImageTimeStampPosX(short XPos)

Function:

Sets Image TimeStamp X location.

Argument:

short XPos X coord.

Return Value:

none

Remarks: See SetImageTimeStampData()

GetImageTimeStampPosX()

Function:
Gets Image TimeStamp X location.
Argument:
none
Return Value:
short X coord .
Remarks:
See SetImageTimeStampData()

SetImageTimeStampPosY(short YPos)

Function:
Sets Image TimeStamp Y location.
Argument:
short YPos Y coord.
Return Value:
none
Remarks:
See SetImageTimeStampData()

GetImageTimeStampPosY()

Function:
Gets Image TimeStamp Y location.
Argument:
none
Return Value:
short Y coord .
Remarks:
See SetImageTimeStampData()

SetImageTimeStampSize(short Size)

Function:
Sets Image TimeStamp size.
Argument:
short Size Font size.
Return Value:
none
Remarks:
See SetImageTimeStampData()

GetImageTimeStampSize()

Function:
Gets Image TimeStamp Size.
Argument:
none
Return Value:
short Font size.
Remarks:
See SetImageTimeStampData()

SetImageXSize(Xsize)

Function:
Sets the current Image file width in pixels.
Argument:
short integer number of pixels desired in the image file. Defaults to 0, which links image size to equal LogicalXSize value.
Return Value:
none.
Remarks:
Maximum useful resolution is based on 410 points per inch of tablet active area.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

Minimum useful resolution is based upon trade-off of desired image coarseness versus resulting image file size.

GetImageXSize

Function:

Gets the current Image file width in pixels.

Argument:

none

Return Value:

short integer value of image file X size. If zero, then value is as specified by the LogicalXSize.

Remarks:

The currently set image file width is a decimal integer in pixels.

SetImageYSize(Ysize)

Function:

Sets the current Image file height in pixels.

Argument:

short integer number of pixels desired in the image file. Defaults to 0, which links image size to equal LogicalYSize value.

Return Value:

none.

Remarks:

Maximum useful resolution is based on 410 points per inch of tablet active area. Minimum useful resolution is based upon trade-off of desired image coarseness versus resulting image file size.

GetImageYSize()

Function:

Gets the current Image file height in pixels.

Argument:

none

Return Value:

short integer value of image file Y size. If zero, then value is as specified by the LogicalYSize.

Remarks:

The currently set image file width is a decimal integer in pixels.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com

Statement regarding patents, and warning against modification of Topaz products:
Topaz products are covered under US patents 6,307,955 and 5,120,908 and 5,122,623. Patent work is ongoing. Use of Topaz's products in accordance with our instructions, to the best of our knowledge, does not infringe any patents. However, be aware that there are many patents out there, and modification of our products, or use of our products for other than their intended purpose, could run afoul of these patents. It is the responsibility of Topaz customers to honor relevant patents. U. S. Patent Nos. 5,120,906; 5,195,133; 5,227,590; 5,297,202; 5,322,978; 5,544,255; 5,647,017; 5,818,955; 6,064,751; 6,091,835; 6,381,344; 6,539,363 and others are patents that users of Topaz products should consider if modifications are to be made to our products or if our products are to be used for other than their intended purpose. These patents may be obtained at http://www.uspto.gov.Only as examples, without first considering these patents, you should refrain from storing within the signature information (sig file or sig data):
1. A reason for signing or a statement of importance of the signature together with a set of measurements relating to the handwritten signature and either an indication of the signatory or means for comparing said measurements with a set of statistics of a genuine signature to obtain a similarity score.
2. A visual representation of a signature together with a document checksum, hash, or receipt, and the claimed identity of the signatory.
3. A set of measurements relating to biometric information together with a document checksum, hash, or receipt, and the claimed identity of the user.
The foregoing statement is not intended to be an interpretation of the patents or their application to any particular product or implementation.
It may be possible that Federal (FDA) and state digital signature regulations become violated if techniques in 1, 2, and 3 above are employed, so rest assured, Topaz software does not do any of them. Note the information below:
1. The FDA regulations state that "Signed electronic records shall contain information associated with the signing that clearly indicates all of the following ...printed name ... date and time ... The meaning ...". (see FDA guidelines section 11.50) The FDA regulations treat electronic records and electronic signatures as different entities (Section 11.70). Therefore, you should refrain from modifying the Topaz system on your own to store this document-related data in the signature file.
2. State regulations such as CA code of Regulations, Title 2, Division 7, Chapter 10 for example, require that the signature be bound to only the single message that is signed and not to any other message. Therfore you should refrain from modifying the Topaz system on your own to place a checksum in with the signature data and then adding other information in with the signature, and then binding them to a secret key. If your were to modify the Topaz system to do this, you would be binding the signature to multiple-messages which is prohibited by the state regulations.
3. State regulations require that the signature data be capable of verification by a forensic document examiner. (CA regulations paragraph 220003(b)(3)(B) for example). Some of the techniques in 1-3 above appear at odds with "lengthy process of handwriting analysis" required by state regulation (see state of CA FAQ section). With the Topaz system, you are assured of the most accurate forensic handwriting analysis results, since the Topaz .sig file is saving all of the original signature data.
In addition, there are potential security and refutability issues if too many things are crammed in with the signature. Leave the document data, reason for signing, receipt, and identity information in the document as SigPlus is designed to work, and as complies with FDA and state regulations. That way, the signature is bound to all of the data, not just to part of the data. It is very easy to create all kinds of standard image files of the signature using SigPlus software. Please contact the factory if you have any additional questions or would like more information about your rights concerning patents as a customer or developer using Topaz products.

Important Notice:

This software or any or all additional documentation, guidelines, or examples do not constitute a warranty about the performance, security, or legal acceptability of SigPlus software or the SigPlus control in any specific use or implementation. To the extent that SigPlus or SigPlus are used to achieve regulatory or other specific objectives within an industry, you must consult competent experts or regulatory officials together with your own plan to achieve your desired business objectives using the Topaz tools.

Topaz Systems, Inc., 650 Cochran Street, Unit 6, Simi Valley, CA, USA, 93065
Phone: 805 520-8282 Fax: 805 520-0867 www.topazsystems.com