Snagit®

COM Server Documentation

Release 11.0.0

February 2012

TechSmith
Show the world.

Table of Contents

Introduction to the Snagit COM Server

Using Snagit's COM Server, you can easily incorporate Snagit's screen capture functionality into your organization's proprietary applications. The COM Server lets you customize capture instructions for Snagit, automate the capture process, and programmatically control Snagit's entire array of image and text capture features from within your own application.

Snagit's add-ins for Microsoft and Adobe products – which allow you to launch Snagit from the toolbar within these applications – are great examples of the power and convenience of Snagit's COM Server.

You can access Snagit's capture functionality from any programming language that supports COM, including the following:

- Visual Basic
- VBScript
- Visual C/C++
- Visual Studio .NET C#
- Delphi
- C++Builder

This Snagit COM Server User Guide assumes you are familiar with COM programming. If you need more information about COM, please consult your development environment's documentation.

Installation Information

The Snagit COM Server is built into Snagit. Download and install Snagit. This automatically registers the Snagit COM Server and makes it available to other applications.

The Snagit COM Server is implemented as an out-of-process COM Server. The Snagit setup program registers the Snagit COM Server after you install Snagit. If you need to manually register the COM Server, use a command prompt and type the following from the Snagit program folder:

```
Snagit32.exe /register
```

Application Deployment and Licensing

Each desktop using the Snagit COM Server must have Snagit installed. Any commercially distributed software application using the Snagit COM Server must comply with the restrictions found in the Snagit Software License Agreement found in the license.txt file in the Snagit application folder.

> If you are using the Snagit evaluation version, the COM Server is fully functional during the 30-day evaluation period. Once the 30-day period has expired, invoking the capture method fails, and the COM Server gives a **serrSnagitExpired** error, and **LastError** is set to **serrSnagitExpired**. All programs using the Snagit COM Server should check the **LastError** property or subscribe to the events from the capture interfaces, so that this error case is handled correctly.

System Requirements

An application using the Snagit COM Server must meet the following system requirements:

- Microsoft Windows XP*, Windows Vista*, or Windows 7*
- 60 MB of hard-disk space for program installation
- Internet Explorer version 7.0 or later
- 1.0 GHz processor ~ Recommended: 2.4 GHz
- 512 MB of RAM ~ Recommended: 1 GB or more

* 32-bit or 64-bit

What's New

The following changes have been made for the most recent version of Snagit.

Version 11

As of Snagit version 11, the following changes have been made:

- Some enumerated types have had values removed.
  - snagImageInput
    - siiDOSScreen
    - siiProgramFile
    - siiEllipse
    - siiRoundedRect
    - siiTriangle
    - siiPolygon
    - siiWallpaper
    - siiDirectX
  - snagTextInput
    - stiDOSScreen
  - snagImageFileType
    - siftPCX
    - siftMSP
    - siftCAL
    - siftGEM
    - siftIOCAR
    - siftIOCAW
    - siftPICT
    - siftLEAD
  - snagColorConversionMethod
    - sccmCustom = 4
- Other enumerated types were removed entirely
  - snagDitherPalette
  - snagDitherMethod

- Color depth filters were removed
  - IImageColorConversion properties
    - snagImageColorDepth
    - snagDitherMethod
    - snagDitherPalette

Version 10

As of Snagit version 10, the following features have been added:

- New image input setting, **siiCapture**, for All-in-One Capture ®.
- **ITextCapture2** and **ITextFile2** interfaces added for RTF text capture support.

Getting Started

The Snagit COM Server should be accessible from any language that supports COM. Only basic data types are used, see **Data Types** for more information.

> The Snagit COM Server is not an automation server. None of the changes made through the COM Server are saved in the Snagit application or the registry. You can use the COM Server from multiple clients, so check return values on calls to methods in case a conflict has occurred.

How It Works

All the interfaces, properties, and methods described in this user guide are accessible from any COM enabled language.All the interfaces, properties, and methods described in this user guide are accessible from any COM enabled language.

Interfaces	These define the properties, methods, and events for a given COM object.
Properties	Values stored in an interface that may be set or retrieved. Some properties are read only; all interfaces are read only.
Methods	Functions in interfaces that interact with the COM server.
Events	Events may be sent from the COM server to the client program. The client program must “subscribe” to the event interface to receive the events. This is a language dependent option, and not all programming languages can receive events.
Enumerations	There are many enumerations used in the Snagit COM server. These are described individually in the **Enumerations** section.

To access the COM object’s functionality, you must first create an instance of the object in your client program. See the tutorial for an example using Visual Basic 6.

- **C++ MFC and ATL Users**

  The easiest way to make use of the COM server is using the Microsoft Visual C++ #import statement on the Snagit.tlb file present in the Snagit install directory. This creates smart-pointer wrapper classes for the COM server.

- **Visual Basic 6 Users**

  A reference to the SNAGIT 1.0 Type Lib should be added in the *References* dialog accessed from the **Project > References** menu item.

- **VB.NET and C# Users**

  Choose **Project > Add Reference** to add a reference to the SNAGIT 1.0 Type Lib. Note that the entry is on the COM tab page.

Image Capture Code Tutorial

This section gives you step-by-step instructions on the basic methods and properties you need to create an image capture. The source code fragments are given using Microsoft's Visual Basic programming language.

See the section on Visual Basic in Language Specific Samples for the complete source code showing how to use the COM object. View additional samples in the following programming languages:

- C++/ATL
- C#
- VB.NET
- VBScript

Step 1 – Creating the Capture Object

Create the capture object.

```
Dim ImageCapture1 As SNAGITLib.ImageCapture

Set ImageCapture1 = CreateObject("Snagit.ImageCapture.1")
```

Step 2 – Setting the Capture Properties

Configure various capture object properties. All object properties use default values if the property is not specified. Configure different properties based on what you want to do with the capture object.

```
// Configure to capture the a window and save to the clipboard

ImageCapture1.Input = siiWindow

ImageCapture1.Output = sioClipboard
```

If you do not know the current capture setting, you can retrieve these values.

```
Dim bUsePreview

bUsePreview = ImageCapture1.EnablePreviewWindow

Dim CaptureCursor CaptureCursor = ImageCapture1.IncludeCursor
```

Step 3 – Starting the Capture

Once the capture object is configured correctly, start the capture session.

```
ImageCapture1.Capture
```

> The Capture call is asynchronous. The capture object may still be completing the operation after the function returns.

Step 4 – Getting Capture Information

During or after a capture, you may want to monitor the state of the capture.

```
Dim capState As snagCaptureState

capState = ImageCapture1.CaptureState

Dim capDone

capDone = ImageCapture1.IsCaptureDone
```

The capture state information resets when a new capture is started.

> Specific interfaces send event callbacks to which a client can subscribe. This allows the client program to update without polling.

Interfaces

Interfaces define the methods and properties of the COM object.

When available, methods are listed before properties on interface pages.

Properties can be write or read only. Any property that is used to access another interface is read only.

The following interfaces are available:

- IAutoScrollOptions
- IClipboardOptions
- IDelayOptions
- IExtendedWindowOptions
- IFTPOptions
- IImageAnnotation
- IImageBorder
- IImageCaptionOptions
- IImageCapture
- IImageCapture2
- IImageCaptureResults
- IImageColorConversion
- IImageColorEffects
- IImageColorSubstitution
- IImageFile
- IImageFile2
- IImageFilters
- IImageResolution
- IImageScale
- IImageTrim
- IImageWatermark
- IMailOptions
- IMenuOptions
- IObjectTextOptions
- IPrinterOptions
- IPrinterPageLayoutOptions
- IRegionOptions
- ISelectedArea
- ITextAnnotation
- ITextCapture
- ITextCapture2
- ITextFile
- ITextFile2
- ITextFilters
- ITextFont
- ITextLayout

- ITWAINOptions
- IWindowOptions

IAutoScrollOptions

The **IAutoScrollOptions** interface contains the settings for scrolling captures. These settings are only valid when the input of the capture object is set to **siiWindow** or **siiRegion** on an image capture object and **stoWindow** or **stoRegion** on a text capture object.

Properties

Property (type)	Default	Read / Write	Notes
AutoScrollMethod (snagAutoScrollMethod)	sasmNone	Write	Sets the type of automatic scrolling for the next capture. See snagAutoScrollMethod for information on possible options.
ForegroundScrollingWindow (Boolean)	True	Write	If enabled, the next capture performed brings the window to be scrolled to the foreground and displays the window while scrolling.
Delay (long)	0	Write	The time, in seconds, to wait between foregrounding the window and capturing. Valid values are 0 to 300000.
StartingPosition (snagAutoScrollStartingPosition)	sasspTopLeft	Write	The position from which the window is scrolled. For more information, see snagAutoScrollStartingPosition.

IClipboardOptions

The **IClipboardOptions** interface is used to set the options related to Clipboard capture input.

Properties

Property (type)	Default	Read / Write	Notes
WidthInPixels (long)	640	Write	Controls the width (in pixels) of Clipboard capture images. Set this value to zero to use the Windows desktop width and height for image dimensions.

IDelayOptions

The **IDelayOptions** interface specifies the number of seconds to wait before taking a capture.

Properties

Property (type)	Default	Read / Write	Notes
EnableDelayedCapture (Boolean)	False	Write	If set to true, the Snagit COM server waits for the period of time specified in the **DelaySeconds** property before performing the capture.
DelaySeconds (long)	10	Write	Specifies the number of seconds to delay before performing a capture. Ignored if **EnableDelayedCapture** is set to false.
EnableCountdownWindow (Boolean)	True	Write	If set to true, a countdown is displayed showing the number of seconds left before the capture occurs. Ignored if **EnableDelayedCapture** is set to false.

IExtendedWindowOptions

The **IExtendedWindowOptions** interface contains all the settings for an extended window capture. Extended window capture allows you to resize a window to any width/height before capturing (potentially bigger than the screen). The window is restored to its original width/height after capture is completed. You can also use this option to capture a window partially off the screen.

Only valid when capturing an image object using the **siiExtendedWindow** input.

Properties

Property (type)	Default	Read / Write	Notes
EnablePreview (Boolean)	True	Write	If true, the user sets the height and width for the extended window capture. Ignored if **UseSpecifiedCaptureSize** is false.
Height (long)	600	Write	The height to size the window before taking the capture. Must be a positive integer.
UseSpecifiedCaptureSize (Boolean)	True	Write	
Width (long)	800	Write	The width to size the window before taking the capture. Must be a positive integer.

IFTPOptions

The **IFTPOptions** interface is used when the output is set to **sioFTP** or **stoFTP**. The capture image or text is sent to an FTP server. At a minimum, the server and filename must be set in order to use FTP output.

Properties

Property (type)	Default	Read / Write	Notes
EnableProgressDialog (Boolean)	False	Write	When true, displays a dialog that detailing the status of the current transfer.
Filename (string)	"" (empty string)	Write	File name to use if **UseAutomaticFileNaming** is false, or the prefix of the file name if **UseAutomaticFileNaming** is true.
Password (string)	"" (empty string)	Write	Password on the FTP server, if it requires authentication. Use with **ServerRequiresAuthentication**.
Port (long)	21	Write	The port of the FTP server to upload to.
ProxyServer (string)	"" (empty string)	Write	Specifies both the proxy server name and the port in the format "server:port". Only used if **UseProxyServer** is true.
RemotePath (string)	"" (empty string)	Write	Path to a directory on the remote server to put the output file. This can be relative or absolute, but must exist.
SequenceLimit (long)	0	Write	Limits the number of auto-named files stored on the FTP server (oldest file is deleted). Use 0 for no limit.
Server (string)	"" (empty string)	Write	Location of the FTP server to upload to. This does not need the "ftp://" prefix.
ServerRequiresAuthentication (Boolean)	False	Write	Use with **UserName** and **Password**.
UseAutomaticFileNaming (Boolean)	False	Write	When true, the file is automatically named. Numbers are appended to the end of the file name , starting with 1. Files of separate types (extensions) have separate numbering.
UsePassiveFTP (Boolean)	False	Write	Enable or disable passive FTP mode.
UseProxyServer (Boolean)	False	Write	Use when routing transfers through a proxy server.
UserName (string)	"" (empty string)	Write	User name on the FTP server, if it requires authentication. Use with **ServerRequiresAuthentication**.

Property (type)	Default	Read / Write	Notes
EnableProgressDialog (Boolean)	False	Write	When true, displays a dialog that detailing the status of the current transfer.
UseTempFileReplace (Boolean)	True	Write	When true, the is uploaded under a temporary name and then renamed to the appropriate output file name.

IImageAnnotation

The **IImageAnnotation** interface allows for overlaying user or system captions on the captured image. The length of the string is limited by width of the image or by the width of the printed page. The text is truncated if it is too long.

Properties

Property (type)	Default	Read / Write	Notes
EnableCaption (Boolean)	False	Write	Enables the text caption.
PromptForCaption (Boolean)	False	Write	Prompts the user for caption text. Ignored if **EnableCaption** is set to false.
CaptionText (string)	"" (empty string)	Write	The text use in the caption.
IncludeComputerName (Boolean)	False	Write	Appends the name of the computer to the caption.
IncludeUserName (Boolean)	False	Write	Appends the name of the current user to the caption.
IncludeTimeDate (Boolean)	False	Write	Appends the time and/or date to the system caption.
TimeDateDisplayOrder (**snagTimeDateOrder**)	stdoTimeThenDate	Write	Defines if, and in what order, to display the time and date.
UseWindowsTimeFormat (Boolean)	True	Write	Use current Windows standard format for time.
UseWindowsDateFormat (Boolean)	True	Write	Use current Windows standard format for date.
CustomTimeFormat (string)	"" (empty string)	Write	Text representing the format of the time, such as "h:mm:ss tt".
CustomDateFormat (string)	"" (empty string)	Write	Text representing the format of the date, such as "M/d/yyyy".
PrintCaptionsAtPageBottom (Boolean)	True	Write	If true, the caption is not printed over the image, but instead added to the bottom.
CaptionOptions (**IImageCaptionOptions**)		Read	An interface containing settings for the text displayed as a caption.
SystemCaptionOptions (**IImageCaptionOptions**)		Read	An interface containing settings for the text displayed as a caption.

IImageBorder

The **IImageBorder** interface is a filter that creates a border around the edge of an image. This filter may increase the size of the original captured image.

Properties

Property (type)	Default	Read / Write	Notes
EnableBorder (Boolean)	False	Write	Enables the border filter.
FrameOutsideImage (Boolean)	True	Write	If true, draws the border outside the bounds of the captured image.
Use3DEffect (Boolean)	True	Write	Uses three colors to create a 3D effect.
TotalWidth (long)	8	Write	The total width of the border, in pixels.
ShadowWidth (long)	2	Write	The width, in pixels, of the shadow effect. Valid values are 0 to one half of the value of **TotalWidth**.
MainColor (long)	RGB (192, 192, 192)	Write	The default border color. Any RGB color is a valid value. This is the only color used if **Use3DEffect** is false.
HighlightColor (long)	RGB (255, 255, 255)	Write	The highlight color of the 3D border effect. Any RGB color is a valid value. This color is used only if **Use3DEffect** is true.
ShadowColor (long)	RGB (128, 128, 128)	Write	The shadow color of the 3D border effect. Any RGB color is a valid value. This color is used only if **Use3DEffect** is true.

For more information on RGB color values, see **Colors**.

IImageCaptionOptions

The **IImageCaptionOptions** interface contains the settings for the text that is added as a caption to image captures. Use **IImageAnnotation** to turn add captions to image captures.

Properties

Property (type)	Default	Read / Write	Notes
BackgroundColor (long)	RGB (0, 0, 0)	Write	The background color of the rectangle on which the text is displayed. Any RGB color is a valid value. If **UseTransparentBackground** is true, this property is not used.
CaptionStyle (**snagCaptionTextStyle**)	sctsNormal	Write	Text style for the caption.
Font (**ISelectedArea**)		Read	An interface that exposes the bounding box around the captured area.
OutlineColor (long)	RGB (255, 255, 255)	Write	The color of the outline that surrounds the text. Any RGB color is a valid value.
Placement (**snagPlacement**)	spCenterTop	Write	Where the text is placed on the captured image.
ShadowColor (long)	RGB (192, 192, 192)	Write	The color of the drop shadow effect for the text. Any RGB color is a valid value.
TextColor (long)	RGB (0, 0, 0)	Write	The font color. Any RGB color is a valid value.
UseTransparentBackground (Boolean)	False	Write	Hides the rectangle on which the text is displayed.
UseWordWrap (Boolean)	True	Write	Wraps the text to a new line if the text is wider than the display area.

> For more information on RGB color values, see **Colors**.

IImageCapture

The **IImageCapture** interface is the base capture interface for image captures. All image capture actions and settings are contained within it.

> If your users have Snagit version 8.1.0 or later, use the **IImageCapture2** interface for image captures.

Methods

Name	Action	Notes
Capture	Invokes an image capture	Moves the Snagit COM server into a busy state while the capture is performed. This is always non-blocking except when the output mode is set to **sioFile** with the Editor window disabled. In order to know when a capture finishes, subscribe to either the **_ICaptureEvents** or poll the **IsCaptureDone** property.

Properties

Property (type)	Default	Read/Write	Notes
AutoScrollOptions (IAutoScrollOptions)		Read-only	An interface containing the settings for scrolling captures.
BackgroundColor (Long)	RGB(255, 255, 255)	Write	Sets the background color to use outside the capture area for the next capture performed. Any RGB color value is valid (see Colors).
CaptureMultipleAreas (Boolean)	False	Write	Enables or disables multiple area selection for the next capture performed.
CaptureState (snagCaptureState)	scsIdle	Read-only	Returns the current capture state set by the Snagit COM server. Instead of subscribing to the event interface, you can poll this property for the state of the Snagit COM server.
ClipboardOptions (IClipboardOptions)		Read-only	An interface to set options related to Clipboard capturing inputs.
DelayOptions (IDelayOptions)		Read-only	An interface that allows for a capture to be called after a specified number of seconds.
EnablePreviewWindow (Boolean)	False	Write	Enables or disables the Editor window for the next capture.
Filters (IImageFilters)		Read-only	A collection of interfaces for adjusting a captured image.
ForegroundPreview (Boolean)	True	Write	When enabled, this forces the Editor window into the foreground (gives it focus) after making a capture. Only valid if **EnablePreviewWindow** is set to true.
IncludeCursor (Boolean)	True	Write	Enables or disables the inclusion of the mouse cursor for the next capture performed.
Input (snagImageInput)	siiWindow	Write	Changes the input mode for the next capture performed. The input mode determines where the Sna-

Property (type)	Default	Read/Write	Notes
AutoScrollOptions (**IAutoScrollOptions**)		Read-only	An interface containing the settings for scrolling captures.
			git COM server captures the bitmap from.
InputExtendedWindowOptions (**IExtendedWindowOptions**)		Read-only	An interface containing all settings for an Extended Window capture.
InputMenuOptions (**IMenuOptions**)		Read-only	An interface containing all settings for a Menu capture (input set to **siiMenu**).
InputRegionOptions (**IRegionOptions**)		Read-only	An interface containing all the settings for a Region capture (input set to **siiRegion** or **stiRegion**).
InputTWAINOptions (**ITWAINOptions**)		Read-only	An interface containing all the settings for a TWAIN capture (input mode set to **siiTwain**).
InputWindowOptions (**IWindowOptions**)		Read-only	An interface containing the settings for a Window capture (input set to **siiWindow** or **stiWindow**).
IsCaptureDone (Boolean)	False	Read-only	After a call to the **Capture** method has been made, this property tests to see if the capture has been completed. This property blocks and performs a polling to see when the capture state goes back to **scsIdle**.
LastCaptureSucceeded (Boolean)	True	Read-only	Returns true if the last capture attempt was successful. Returns false if the last capture attempt was unsuccessful.
LastError (**snagError**)	serrNone	Read-only	Holds the last error set in the Snagit COM server. If an application subscribes to the **OnStateChange** event of the **_ICaptureEvents** interface, this property can be read for the error on a **scsCaptureFailed** state change. This property is reset to **serrNone** when a new call to the **Capture** method is made.
LastFileWritten (string)	"" (empty	Read-only	If the last capture used **sioFile** as

Property (type)	Default	Read/Write	Notes
AutoScrollOptions (**IAutoScrollOptions**)		Read-only	An interface containing the settings for scrolling captures.
	string)		the output, this property contains the whole path for the resulting image file.
NotificationType (**snagNotificationType**)	sntNone	Write	Sets the messages the Snagit COM server displays. For more information on notifications see **snagNotificationType** enumeration description.
Output (**snagImageOutput**)	sioFile	Write	Changes the output mode for the next capture performed. The output mode determines what the Snagit COM server does with the resulting bitmap from the capture.
OutputFTPOptions (**IFTPOptions**)		Read-only	An interface containing settings for sending the capture out to FTP (output set to **sioFTP** or **stoFTP**). At a minimum, server and filename must be set.
OutputImageFile (**IImageFile**)		Read-only	An interface for the configuration of an output image file.
OutputMailOptions (**IMailOptions**)		Read-only	An interface for setting email output information. Either all information is specified and the email is sent immediately, or the user is prompted for all information.
OutputPrinterOptions (**IPrinterPageLayoutOptions**)		Read-only	An interface used to specify the formatting of an output being sent to a printer (output set to **sioPrinter** or **stoPrinter**).
UseMagnifierWindow (Boolean)	True	Write	When enabled, a magnifier window shows next to the capture crosshairs for precision when defining capture regions.

IImageCapture2

The **IImageCapture** interface is the base capture interface for image captures. All image capture actions and settings are contained within it.

> **IImageCapture2** replaces **IImageCapture** for users of versions 8.1.0 and later.

Methods

Name	Action	Notes
Capture	Invokes an image capture	Moves the Snagit COM server into a busy state while the capture is performed. This is always non-blocking except when the output mode is set to **sioFile** with the Editor window disabled. In order to know when a capture finishes, subscribe to either **_ICaptureEvents** or poll the **IsCaptureDone** property.

Properties

Property (type)	Default	Read/Write	Notes
AutoScrollOptions (**IAutoScrollOptions**)		Read-only	An interface containing the settings for scrolling captures.
BackgroundColor (long)	RGB(255, 255, 255)	Write	Sets the background color to use outside the capture area for the next capture performed. Any RGB color value is valid (see **Colors**).
CaptureMultipleArea (Boolean)	False	Write	Enables or disables multiple area selection for the next capture performed.
CaptureResults (**IImageCaptureResults**)		Read-only	An interface that stores information about what happened during a capture.
CaptureState (**snagCaptureState**)	scsIdle	Read-only	Returns the current capture state set by the Snagit COM server. Instead of subscribing to the event interface, you can poll this property for the state of the Snagit COM server.
ClipboardOptions (**IClipboardOptions**)		Read-only	An interface to set options related to Clipboard capturing inputs.
DelayOptions (**IDelayOptions**)		Read-only	An interface that allows for a capture to be called after a specified number of seconds.
EnablePreviewWindow (Boolean)	False	Write	Enables or disables the Editor window for the next capture.
Filters (**IImageFilters**)		Read-only	A collection of interfaces for adjusting a captured image.
ForegroundPreviewWindow (Boolean)	True	Write	When enabled, this forces the Editor window into the foreground (gives it focus) after making a capture. Only valid if **Ena-**

Property (type)	Default	Read/Write	Notes
			blePreviewWindow is set to true.
HotspotType (**snagHotspotType**)	shtNone	Write	Changes the type of hotspot captured. Hotspots are not compatible with all capture types.
IncludeCursor (Boolean)	True	Write	Enables or disables the inclusion of the mouse cursor for the next capture performed.
Input (**snagImageInput**)	siiWindow	Write	Changes the input for the next capture. Input determines where the bitmap is captured from.
InputExtendedIWindowOptions (**IExtendedWindowOptions**)		Read-only	An interface containing all settings for an Extended Window capture.
InputMenuOptions (**IMenuOptions**)		Read-only	An interface containing all settings for a Menu capture (input set to **siiMenu**).
InputRegionOptions (**IRegionOutputs**)		Read-only	An interface containing all the settings for a Region capture (input set to **siiRegion** or **stiRegion**).
InputTWAINOptions (**ITWAINOptions**)		Read-only	An interface containing all the settings for a TWAIN capture (input mode set to **siiTwain**).
InputWindowOptions (**IWindowOptions**)		Read-only	An interface containing the settings for a Window capture (input set to **siiWindow** or **stiWindow**).
IsCaptureDone (Boolean)	False	Read-only	After a call to the **Capture** method has been made, this property tests to see if the capture has been completed. This property blocks and performs a polling to see when the capture state goes back to **scsIdle**.
LastCaptureSucceeded (Boolean)	True	Read-only	Returns true if the last cap-

Property (type)	Default	Read/Write	Notes
			ture attempt was successful. Returns false if the last capture attempt was unsuccessful.
LastError (snagError)	serrNone	Read-only	Holds the last error set in the Snagit COM server. If an application subscribes to the **OnStateChange** event of the **_ICaptureEvents** interface, this property can be read for the error on a **scsCaptureFailed** state change. This property is reset to **serrNone** when a new call to the **Capture** method is made.
LastFileWritten (string)	"" (empty string)	Read-only	If the last capture used **sioFile** as the output, this property contains the whole path for the resulting image file.
NotificationType (snagNotificationType)	sntNone	Write	Sets the messages the Snagit COM server displays. For more information on notifications see the **snagNotificationType** enumeration description.
Output (snagImageOutput)	sioFile	Write	Changes the output for the next capture. Output determines what happens to the captured bitmap.
OutputFTPOptions (IFTPOptions)		Read-only	An interface containing settings for sending the capture out to FTP (output set to **sioFTP** or **stoFTP**). At a minimum, server and filename must be set.
OutputImageFile2 (IImageFile2)		Read-only	An interface for the configuration of an output image file.
OutputMailOptions (IMailOptions)		Read-only	An interface for setting email output information. Either all information is specified and the email is sent immediately, or the user is

Property (type)	Default	Read/Write	Notes
			prompted for all information.
OutputPrinterPageLayoutOptions (**IPrinterPageLayoutOptions**)		Read-only	An interface used to specify the formatting of output going to a printer. Only used when **sioPrinter** or **stoPrinter** is selected as the output.
OutputPrinterOptions (**IPrinterOptions**)		Read-only	An interface for use with printer output. If **UseDefault:Printer** is set to true then other values set programmatically by **SetData** or by the user with the **ShowSelectionDialog** are ignored.
UseMagnifierWindow (Boolean)	True	Write	When enabled, a magnifier window shows next to the capture crosshairs for precision when defining capture regions.

IImageCaptureResults

The **IImageCaptureResults** interface contains information about what happened during a capture.

Properties

Property (type)	Default	Read/Write	Notes
LastFileWritten (string)	"" (empty string)	Read-only	The file path for the image file if **sioFile** was used as the output.
SelectedArea (ISelectedArea)		Read-only	An interface that exposes the bounding box around the captured area. For a freehand region, this is the bounding box around the total area (e.g. the leftmost or rightmost point of the selection).

IImageColorConversion

The **IImageColorConversion** interface sets an image's color resolution during capture.

Properties

Property (type)	Default	Read / Write	Notes
ConversionMethod (snag-ColorConversionMethod)	sccmNone	Write	By default, no color conversion occurs. If set to **sccmCustom**, the **ColorDepth** and dither settings are used. If set to **sccmMonochrome**, the **MonochromeThreshold** is used.
MonochromeThreshold (long)	50	Write	A percentage value that determines the quantity of black and white that shows in the output image. Valid values are from 0 -100.

IImageColorEffects

The **IImageColorEffects** interface is a collection of filters that modify the colors of a captured image.Note that these ranges are different than those presented in the Snagit program.

Properties

Property (type)	Default	Read / Write	Notes
Brightness (long)	0	Write	Changes the intensity (brightness) of the image. Valid values are -1000 to 1000.
Contrast (long)	0	Write	Changes the contrast between light and dark portion of the image. Valid values are -1000 to 1000.
Hue (long)	0	Write	Changes the hue of colors in the image (similar to rotating a color wheel to select a different mixture of colors). Valid values are -360 to 360.
Saturation (long)	0	Write	Changes the saturation of colors in the image. Saturation is the quantity of a color in a pixel, from gray at the lowest saturation to rich color in the highest. Valid values are -1000 to 1000.
Gamma (long)	100	Write	Adjusts the intensity of colors in an image by changing the gamma constant used to map the intensity values. Gamma correction changes brightness using a logarithmic scale for visual perception; gamma is a constant used to calculate the progression. Valid values are 1 to 499.

IImageColorSubstitution

The **IImageColorSubstitution** interface allows colors in the captured image to be replaced. Multiple rules for color substitution can be added with the AddColorSub method.

> Call **ClearColorSub** between captures to clear the current list of color substitutions.

Methods

Method	Action	Parameters	Notes
AddColorSub(long, OrigColor, long New-Color, long Accuracy, Boolean SwapColors)	Adds a color substitution to the list.	[in] **OrigColor** (long): The color to be replaced. [in] **NewColor** (long): The color to replace the original color. [in] **Accuracy** (long): How close the color should be to the color being replaced. [in] **SwapColors** (Boolean): If true, colors are swapped instead of replaced.	Colors are selected using RGB values.
ClearColorSub( )	Clears the current list of color sub-stitutions.		
ShowColorSubDialog (long IParentHandle, Boolean dialogAccepted)	Immediately shows the color sub-stitution dialog.	[in] **IParentHandle** (long): Handle to window (hwnd) in 32-bit integer for-mat. If the handle is not available, set to 0. [out] **dialogAccepted** (Boolean): True if the user selects OK to select a new color substitution. False if the user selects Cancel.	

Properties

Property (type)	Default	Read / Write	Notes
ColorSubMethod (**snag-ColorSubMethod**)	scsmNone	Write	Used to substitute colors for other colors.

For more information on RGB colors, see **Colors**.

IImageFile

The **IImageFile** interface allows for the configuration of an output image file. Always call **LoadImageDefaults** after changing the **FileType** property.

Methods

Method	Action	Parameters
LoadImageDefaults (**snag-ImageFileType**, newFileType)	Sets the file type and loads the default attributes for that type (subtype, quality, etc).	[in] newFileType (**snag-ImageFileType**): The file type to load default settings for.
ShowFileSettingDialog (long lParentHandle, **snag-ImageFileType**, newFileType, Boolean dialogAccepted)	Shows the dialog for a given file type and sets the file type if the user accepts the dialog.	[in] lParentHandle (long): Handle to windows (hwnd) in 32-bit integer format. If handle is not available, set value to 0. [out] dialogAccepted (Boolean): True if the user clicked OK to set new file type settings. False if the user clicked Cancel.

Properties

Property (type)	Default	Read / Write	Notes
AutoFilePrefix (string)	"SNAG-"	Write	When the **FileNamingMethod** is set to **sofnmAuto**, this string is the prefix used to automatically name files.
AutoNumPrefixDigits (long)	4	Write	When **FileNamingMethod** is set to **sofnmAuto**, this property specifies the number of digits to include in the automatically named output file.
ColorDepth (**snagImageColorDepth**)	sicdAuto	Write	Number of bits used for color encoding.
Directory (string)	"" (empty string)	Write	Sets the directory for the output file. If the **FileNamingMethod** is not set to **sofnmPrompt**, the Snagit COM server fails if this property is not set.
Filename (string)	"SNAGIT"	Write	Specifies the output filename to use. This is only used if **FileNamingMethod** is set to **sofnmFixed**.
FileNamingMethod (**snagOutputFileNamingMethod**)	sofnmPrompt	Write	Method for naming the output file.
FileType (**snagImageFileType**)	siftBMP - Windows Bitmap	Write	The format of the output file. Use **LoadImageDefaults** to avoid unexpected results.
FileSubType (**snagImageFileSubType**)	-indeterminate-	Write	Further specifies the type of image file to save. Do not assume a default subtype; instead choose a compatible subtype, call **LoadImageDefaults**, or use the **ShowFileSettingDialog** function.
ProgressiveOption (long)	0 - 12	Write	Sets the number of scans for a progressively encoded image. A value of 0 means no progressive scan. This property is only valid is the **FileType** is set to **siftJPEG**.

Property (type)	Default	Read / Write	Notes
Quality (long)	25	Write	Sets the quality of the compression on the image file. This property is only valid if the **FileType** property is set to **siftJPEG**. Valid values are 1 - 100.

IImageFile2

The **IImageFile** interface allows for the configuration of an output image file. Always call **LoadImageDefaults** after changing the **FileType** property.

Methods

Method	Action	Parameters
LoadImageDefaults (snagImageFileType, newFileType)	Sets the file type and loads the default attributes for that type (subtype, quality, etc).	[in] newFileType (**snagImageFileType**): The file type to load default settings for.
ShowFileSettingDialog (long IParentHandle, snagImageFileType, newFileType, Boolean dialogAccepted)	Shows the dialog for a given file type and sets the file type if the user accepts the dialog.	[in] IParentHandle (long): Handle to windows (hwnd) in 32-bit integer format. If handle is not available, set value to 0. [out] dialogAccepted (Boolean): True if the user clicked OK to set new file type settings. False if the user clicked Cancel.

Properties

Property (type)	Default	Read / Write	Notes
AutoFilePrefix (string)	"SNAG-"	Write	When the **FileNamingMethod** is set to **sofnmAuto**, this string is the prefix used to automatically name files.
AutoNumPrefixDigits (long)	4	Write	When **FileNamingMethod** is set to **sofnmAuto**, this property specifies the number of digits to include in the automatically named output file.
ColorDepth (**snagImageColorDepth**)	sicdAuto	Write	Number of bits used for color encoding.
Directory (string)	"" (empty string)	Write	Sets the directory for the output file. If the **FileNamingMethod** is not set to **sofnmPrompt**, the Snagit COM server fails if this property is not set.
Filename (string)	"SNAGIT"	Write	Specifies the output filename to use. This is only used if **FileNamingMethod** is set to **sofnmFixed**.
FileNamingMethod (**snagOutputFileNamingMethod**)	sofnmPrompt	Write	Method for naming the output file.
FileSubType (**snagImageFileSubType**)	-indeterminate-	Write	Further specifies the type of image file to save. Do not assume a default subtype; instead choose a compatible subtype, call **LoadImageDefaults**, or use the **ShowFileSettingDialog** function.
FileType (**snagImageFileType**)	siftBMP - Windows Bitmap	Write	The format of the output file. Use **LoadImageDefaults** to avoid unexpected results.
ProgressiveOption (long)	0 - 12	Write	Sets the number of scans for a progressively encoded image. A value of 0 means no progressive scan. This property is only valid is the **FileType** is set to **siftJPEG**.

Property (type)	Default	Read / Write	Notes
Quality (long)	25	Write	Sets the quality of the compression on the image file. This property is only valid if the **FileType** property is set to **siftJPEG**. Valid values are 1 - 100.
TransparentColorForGIF (long)	RGB (255,255,255)	Write	The color to make transparent in the GIF file. Valid only when saving GIF files with UseGIFTransparency is set to True. Any RGB color value is valid. See **Colors** for more information.
UseGIFTransparency (Boolean)	False	Write	Sets whether GIF files are saved with a transparent color. The transparent color is set with **TransparentColorForGIF**.

IImageFilters

The **IImageFilters** interface is a collection of interfaces for adjusted captured images. Most filters can be used in conjunction if all choices are compatible.

Properties

Property (type)	Notes
Annotation (**IImageAnnotation**)	An interface for adding annotations to image captures.
Border (**IImageBorder**)	An interface for adding borders to image captures.
ColorConversion (**IImageColorConversion**)	An interface to convert colors in image captures.
ColorEffects (**IImageColorEffects**)	An interface for adding color effects to image captures.
ColorSubstitution (**IImageColorSubstitution**)	An interface for substituting colors in image captures.
Resolution (**IImageResolution**)	An interface for changing the resolution of image captures.
Scale (**IImageScale**)	An interface for resizing image captures.
Trim (**IImageTrim**)	An interface for trimming image captures.
Watermark (**IImageWatermark**)	An interface to add a watermark to image captures.

IImageResolution

The **IImageResolution** interface sets the number of dots per inch (DPI) of the image. These settings have no effect on how the Snagit COM server displays or prints the image.

Properties

Property (type)	Default	Read / Write	Notes
Resolution (long)	96	Write	If **UseAutoResolution** is false, sets the DPI resolution of the image.
UseAutoResolution (Boolean)	True	Write	If true, the Snagit COM server chooses the image resolution. If false, the value specified by **Resolution** is used.

IImageScale

The **IImageScale** interface resizes a captured image.

Rounded, or even proportioned, percentage settings scale faster than irregular ones. If you use a factor that is not proportional (58%, for example), enable **UseSmoothScaling** to improve the image's appearance.

Properties

Property (type)	Default	Read / Write	Notes
Factor (long)	0	Write	Percentage of the original width and height.
FactorHeight (long)	100	Write	Percentage of the original height.
FactorWidth (long)	100	Write	Percentage of the original width.
Height (long)	200	Write	Height, in pixels.
KeepAspectRatio (Boolean)	True	Write	Keeps the image aspect ratio fixed while changing the width or height.
ScaleBy (**snagImageScaleBy**)	sisbFactor	Write	
ScaleMethod (**snagImageScaleMethod**)	sismNone	Write	
UseSmoothScaling (Boolean)	True	Write	This applies a "smoothing" effect images, removing the appearance of jagged edges.
Width (long)	320	Write	Width, in pixels.

IImageTrim

The **IImageTrim** interface removes pixels from the edges of an image. If the **TrimMethod** is set to **stmAuto**, none of the other values in this interface are used.

Properties

Property (type)	Default	Read / Write	Notes
Bottom (long)	0	Write	The number of pixels to trim from the bottom of the image. Only valid if **TrimMethod** is set to **stmManual**.
Left (long)	0	Write	The number of pixels to trim from the left of the image. Only valid if **TrimMethod** is set to **stmManual**.
Right (long)	0	Write	The number of pixels to trim from the right of the image. Only valid if **TrimMethod** is set to **stmManual**.
Top (long)	0	Write	The number of pixels to trim from the top of the image. Only valid if **TrimMethod** is set to **stmManual**.
TrimMethod (**snagTrimMethod**)	stmNone	Write	Determines the method for how pixels are trimmed from an image.

IImageWatermark

The **IImageWatermark** interface is used to create a watermark on a captured image.

Properties

Property (type)	Default	Read / Write	Notes
EmbossDepth (long)	500	Write	Sets the depth of the emboss effect. Higher values created a deeper shadow effect. Valid values are 1-1000.
EmbossDirection (snag-CompassDirection)	scdNorth	Write	Sets the direction of the appearance of the emboss. Ignored if **UseEmboss** is false.
ImageFilename (string)	"" (empty string)	Write	The filename of the image to be used for the watermark.
IncludeWatermark (Boolean)	False	Write	If true, a watermark is added in a fixed location to the image.
KeepAspectRatio (Boolean)	True	Write	If true, scaling performed on the watermark image retains the ratio between the width and height of the watermark source image file.
OffsetHorizontal (long)	10	Write	The amount the watermark is offset horizontally, by a percentage of overall image size, from the placement set in the **Position** property, Valid values are 1 - 100.
OffsetVertical (long)	10	Write	The amount the watermark is offset vertically, by a percentage of overall image size, from the placement set in the **Position** property, Valid values are 1 - 100.
Position (snag-Placement)	spCenterBottom	Write	Sets the position of the watermark in the output image.
Scale (long)	15	Write	The level of scaling (by percentage) done to the watermark image before is added to the capture. Valid values are 1 - 100. In VB6, the "Scale" property is a reserved keyword. Use "ScaleSize" instead. (For Snagit version 8.1.0 or later.)

Property (type)	Default	Read / Write	Notes
EmbossDepth (long)	500	Write	Sets the depth of the emboss effect. Higher values created a deeper shadow effect. Valid values are 1-1000.
EmbossDirection (**snag-CompassDirection**)	scdNorth	Write	Sets the direction of the appearance of the emboss. Ignored if **UseEmboss** is false.
TransparentColor (long)	RGB (128, 128, 128)	Write	Any color in the watermark image matching the RGB value specified is not drawn over the captured image. This property is ignored unless **UseTransparentColor** is set to true.
UseEmboss (Boolean)	False	Write	If true, the watermark image is embossed on the capture.
UseOverlay (Boolean)	False	Write	If true, the watermark is placed on top of the image.
UseSmoothScaling (Boolean)	True	Write	If true, the watermark image is smoothed using interpolation.
UseTransparentColor (Boolean)	False	Write	If true, any colors in the watermark image that are the same as the color in the **TransparentColor** property become transparent in the output image.

IMailOptions

The **IMailOptions** interface sets email output related items. Either all information must be specified and the email is sent immediately, or the user is prompted for all information.

Properties

Property (type)	Default	Read / Write	Notes
Address (string)	"" (empty string)	Write	The email recipient's email address.
MessageText (string)	"" (empty string)	Write	The body text of the email.
Name (string)	"" (empty string)	Write	The email sender's name.
PromptForSettings (Boolean)	False	Write	If false, name, address, subject, and text must be specified programmatically. If true, the user's mail client is shown and the information must be provided by the user. When true, setting other values has no effect.
Subject (string)	"" (empty string)	Write	The subject line of the email.

IMenuOptions

The **IMenuOptions** interface contains all the settings for a menu capture (input set to **siiMenu**).

Properties

Property (type)	Default	Read / Write	Notes
CaptureCascaded (Boolean)	False	Write	When false, only the current menu is included in the capture. When true, the current menu and all parent menus are included in the capture.
IncludeBar (Boolean)	False	Write	Enable or disable inclusion of the menu bar in the capture.

IObjectTextOptions

The **IObjectTextOptions** interface contains all the settings for text object captures. The options are only valid when the input is set to **stoObject**.

Properties

Property (type)	Default	Read / Write	Notes
CaptureDescription (Boolean)	True	Write	When true, the Windows Object Description is included in the text capture. Ignored if **IncludePropertyNames** is set to false.
CaptureName (Boolean)	True	Write	When true, the Windows Object Name is included in the text capture. Ignored if **IncludePropertyNames** is set to false.
CaptureValue (Boolean)	True	Write	When true, the value associated with the object is included in the text capture. Ignored if **IncludePropertyNames** is set to false.
IncludePropertyNames (Boolean)	True	Write	When true, includes **CaptureDescription**, **CaptureName**, and **CaptureValue** if those properties are also true.

IPrinterOptions

The **IPrinterOptions** interface is used with printer output. If **UseDefault:Printer** is set to true then any other values set programmatically by **SetData** or by the user with **ShowSelectionDialog** are ignored.

Methods

Method	Action	Parameters	Notes
GetData (string PrinterName, string DriverName, string Port)	Retrieves data about the selected printer.	[out] PrinterName (string): The name of the selected printer. [out] DriverName (string): The name of the driver installed for the printer. [out] Port (string): The port of the selected printer.	
SetData (string PrinterName, string DriverName, string Port)	Sets the printer for the Snagit capture object to use.	[in] PrinterName (string): The name of the printer to use. [in] DriverName (string): The name of the printer driver to use. [in] Port (string): The port of the selected printer.	If **UseDefault:Printer** is true, data set with this call is overwritten.
ShowSelectionDialog (long IParentHandle, Boolean dialogAccepted)	Displays a printer selection dialog to the user. If a user clicks OK in the dialog, the printer selected in the dialog becomes the selected printer for the Snagit capture object.	[in] IParentHandle (long): Handle to windows (hwnd) in 32-bit integer format. If handle is not available, set value to 0. [out] dialogAccepted (Boolean): True if the user clicked OK to select a new printer. False if the user clicked cancel.	If the user clicks OK on the printer selection dialog, any data set by a call to **SetData** is overwritten. If **UseDefault:Printer** is set to true, any selection made by the user is ignored.

Properties

Property (type)	Default	Read / Write	Notes
UseDefault:Printer (Boolean)	True	Write	If true, the capture is sent to the system default printer. When set to true, this property overrides any other printer settings.

IPrinterPageLayoutOptions

The **IPrinterPageLayoutOptions** interface is used to specify how to format the output going to the printer. These options are only used when either **sioPrinter** or **stoPrinter** is selected as the output for the capture object.

Properties

Property (type)	Default	Read / Write	Notes
Height (long)	4000	Write	The height, in thousandths of an inch, of the capture as it appears on the printed page.
LayoutPosition (**snag-Placement**)	spLeftTop	Write	Sets the placement of the capture as it appears on the printed page.
MarginBottom (long)	0	Write	Sets the minimum distance, in thousandths of an inch, placed between the bottom edge of the printed page and the bottom of the capture.
MarginLeft (long)	0	Write	Sets the minimum distance, in thousandths of an inch, placed between the right edge of the printed page and the right side of the capture.
MarginRight (long)	0	Write	Sets the minimum distance, in thousandths of an inch, placed between the right edge of the printed page and the top of the capture.
MarginTop (long)	0	Write	Sets the minimum distance, in thousandths of an inch, placed between the top edge of the printed page and the top of the capture.
Scale (long)	100	Write	Sets the scale of the capture on the printed page as a percentage.
ScalingType (**snag-PrintScale**)	spsFixedSize	Write	Sets the type of scaling to perform. See **snag-PrintScale** for the available options.
Width (long)	4000	Write	The width, in thousandths of an inch, of the capture as it appears on the printed page.

IRegionOptions

The **IRegionOptions** interface contains all the settings for a region capture (input set to **siiRegion** or **stiRegion**). These options are only valid when using **siiRegion** or **stiRegion** as the input type.

Properties

Properties (type)	Default	Read / Write	Notes
Height (long)	240	Write	Sets the height for the rectangular region to capture. Must be a positive integer.
SelectionMethod (**snagRegionSelectionMethod**)	srsmInteractive	Write	Sets the type of region capture to perform. The default allows for the user to click and drag to select a region.
StartX (long)	0	Write	Sets the starting X coordinate in pixels of the rectangular region to capture. Ignored if **UseStartPosition** is set to false.
StartY (long)	0	Write	Sets the starting Y coordinate in pixels of the rectangular region to capture. Ignored if **UseStartPosition** is set to false.
UseStartPosition (Boolean)	False	Write	When true, the capture is performed with no user input. The capture region is defined by the **Height**, **Width**, **StartX**, and **StartY** properties.
Width (long)	320	Write	Sets the width for the rectangular region to capture. Must be a positive integer.

ISelectedArea

The **ISelectedArea** interface exposes the bounding box around the captured area. For a rectangular region, this is exactly the same. For an irregular region like a freehand area, this is the bounding box around that area (left-most point in the area, right-most, etc).

Properties

Property (type)	Default	Read / Write	Notes
StartX (long)	0	Write	The X coordinate of the upper left corner of the bounding box.
StartY (long)	0	Write	The Y coordinate of the upper left corner of the bounding box.
Height (long)	0	Write	Sets the height, in "ems," to use for the font. The system uses the nearest height to the specified value that does not exceed it.
Weight (long)	0	Write	Sets the weight, or thickness, of the font.

ITextAnnotation

The **ITextAnnotation** interface is used to add a caption to a capture.

Properties

Property (type)	Default	Read / Write	Notes
EnableCaption (Boolean)	False	Write	If true, a text caption is added to the capture.
PromptForCaption (Boolean)	False	Write	Prompts the user for text to use in the caption.
CaptionText (string)	"" (empty string)	Write	The text to use in the caption. If **PromptForCaption** is true, the text in this property is ignored.
IncludeCompuerName (Boolean)	False	Write	Adds the name of the computer to the caption.
IncludeUserName (Boolean)	False	Write	Adds the name of the current user to the caption.
IncludeTimeDate (Boolean)	False	Write	Appends the time and date to the caption.
TimeDateDisplayOrder (**snagTimeDateOrder**)	stdoTimeThenDate	Write	Defines in what order the time and date are displayed.
UserWindowsTimeFormat (Boolean)	True	Write	Use the system's default time format.
UserWindowsDateFormat (Boolean)	False	Write	Use the system's default date format.
CustomeTimeFormat (string)	"" (empty string)	Write	This text represents the format of the time, such as "h:mm:ss tt".
CustomerDateFormat (string)	"" (empty string)	Write	This text represents the format of the date, such as "M/d/yyyy".

ITextCapture

The **ITextCapture** interface is the base capture interface for text captures. All the text capture actions and settings are contained within it except for RTF support.

Methods

Name	Action	Notes
Capture	Invokes a text capture	Moves the Snagit COM server into a busy state while the capture is performed. This is always non-blocking except when the output mode is set to **sioFile** with the Editor window disabled. In order to know when a capture finishes, subscribe to either the **_ICaptureEvents** or poll the **IsCaptureDone** property.

Properties

Property (type)	Default	Read / Write	Notes
AutoScrollOptions (**IAutoScrollOptions**)		Read-only	An interface containing the settings for scrolling captures.
CaptureState (**snagCaptureState**)		Read-only	Returns the current capture state set by the Snagit COM server. Instead of subscribing to the event interface, you can poll this property for the state of the Snagit COM server.
DelayOptions (**IDelayOptions**)		Read-only	An interface that allows for a capture to be called after a specified number of seconds.
EnablePreviewWindow (Boolean)	False	Write	Enables or disables the Editor window for the next capture.
Filter (**IImageFilters**)		Read-only	A collection of interfaces for adjusting captured text.
ForegroundPreview (Boolean)	True	Write	When enabled, this forces the Editor window into the foreground (gives it focus) after making a capture. Only valid if **EnablePreviewWindow** is set to true.
Input (**snagTextInput**)	stiWindow	Write	Changes the input for the next text capture. Input determines where the text is captured from.
InputRegionOptions (**IRegionOptions**)		Read-only	An interface containing all the settings for a Region capture (input set to **siiRegion** or **stiRegion**).
InputObjectTextOptions (**IObjectTextOptions**)		Read-only	An interface containing all settings for text object captures. Only valid when using **stoObject** as the input mode for a text capture.
InputWindowOptions (**IWindowOptions**)		Read-only	An interface containing the settings for a Window capture (input set to **siiWindow** or **stiWindow**).
IsCaptureDone (Boolean)		Read-only	After a call to the **Capture** method has been made, this property tests to see if the capture has been completed. This property blocks and performs a polling to see when the capture state goes back to **scsIdle**.
LastCaptureSucceeded		Read-	Returns true if the last capture attempt was

Property (type)	Default	Read / Write	Notes
(Boolean)		only	successful. Returns false if the last capture attempt was unsuccessful.
LastError (**snagError**)		Read-only	Holds the last error set in the Snagit COM server. If an application subscribes to the **OnStateChange** event of the **_ICaptureEvents** interface, this property can be read for the error on a **scsCaptureFailed** state change. This property is reset to **serrNone** when a new call to the **Capture** method is made.
LastFileWritten (string)		Read-only	If the last capture used **sioFile** as the output, this property contains the whole path for the resulting file.
NotificationType (**snagNotificationType**)	sntNone	Write	Sets the messages the Snagit COM server displays. For more information on notifications see the **snagNotificationType** enumeration description.
Output (**snagTextOutput**)	stoFile	Write	Changes the output for the next text capture. Output determines what happens to the captured text.
OutputFTPOptions (**IFTPOptions**)		Read-only	An interface containing settings for sending the capture out to FTP (output set to **sioFTP** or **stoFTP**). At a minimum, server and filename must be set.
OutputMailOptions (**IMailOptions**)		Read-only	An interface for setting email output information. Either all information is specified and the email is sent immediately, or the user is prompted for all information.
OutputPrinterOptions (**IPrinterOptions**)		Read-only	An interface for use with printer output. If **UseDefault:Printer** is set to true then other values set programmatically by **SetData** or by the user with the **ShowSelectionDialog** are ignored.
UseMagnifierWindow (Boolean)	True	Write	When enabled, a magnifier window shows next to the capture crosshairs for precision when defining capture regions.
TextFileOptions (**ITextFile**)		Read-only	An interface containing all settings for text file output. The file, regardless of the naming method used, has a .txt extension. Only valid when using **stoFile** as the output mode for a

Property (type)	Default	Read / Write	Notes
			text capture.

ITextCapture2

The **ITextCapture2** interface is the base capture interface for text captures with added support for RTF capture. All the text capture actions and settings are contained within it.

Methods

Name	Action	Notes
Capture	Invokes a text capture.	Moves the Snagit COM server into a busy state while the capture is performed. This is always non-blocking except when the output mode is set to **sioFile** with the Editor window disabled. In order to know when a capture finishes, subscribe to either the **_ICaptureEvents** or poll the **IsCaptureDone** property.

Properties

Property (type)	Default	Read / Write	Notes
AutoScrollOptions (**IAutoScrollOptions**)		Read-only	An interface containing the settings for scrolling captures.
DelayOptions (**IDelayOptions**)		Read-only	An interface that allows for a capture to be called after a specified number of seconds.
CaptureState (**snagCaptureState**)		Read-only	Returns the current capture state set by the Snagit COM server. Instead of subscribing to the event interface, you can poll this property for the state of the Snagit COM server.
EnablePreviewWindow (Boolean)	False	Write	Enables or disables the Editor window for the next capture.
Filters (**IImageFilters**)		Read-only	A collection of interfaces for adjusting captured text.
ForegroundPreviewWindow (Boolean)	True	Write	When enabled, this forces the Editor window into the foreground (gives it focus) after making a capture. Only valid if **EnablePreviewWindow** is set to true.
Input (**snagTextInput**)	siiWindow	Write	Changes the input for the next text capture. Input determines where the text is captured from.
InputRegionOptions (**IRegionOptions**)		Read-only	An interface containing all the settings for a Region capture (input set to **siiRegion** or **stiRegion**).
InputObjectTextOptions (**IObjectTextOptions**)		Read-only	An interface containing all settings for text object captures. Only valid when using **stoObject** as the input mode for a text capture.
InputWindowOptions (**IWindowOptions**)		Read-only	An interface containing the settings for a Window capture (input set to **siiWindow** or **stiWindow**).
IsCaptureDone (Boolean)		Read-only	After a call to the **Capture** method has been made, this property tests to see if the capture has been completed. This property blocks and performs a polling to

Property (type)	Default	Read / Write	Notes
			see when the capture state goes back to **scsIdle**.
LastError (snagError)		Read-only	Holds the last error set in the Snagit COM server. If an application subscribes to the **OnStateChange** event of the **_ICaptureEvents** interface, this property can be read for the error on a **scsCaptureFailed** state change. This property is reset to **serrNone** when a new call to the **Capture** method is made.
LastCaptureSucceeded (Boolean)		Read-only	Returns true if the last capture attempt was successful. Returns false if the last capture attempt was unsuccessful.
LastFileWritten (string)		Read-only	If the last capture used **sioFile** as the output, this property contains the whole path for the resulting file.
NotificationType (snagNotificationType)	sntNone	Write	Sets the messages the Snagit COM server displays. For more information on notifications see the **snagNotificationType** enumeration description.
Output (snagTextOutput)	stoFile	Write	Changes the output for the next text capture. Output determines what happens to the captured text.
OutputFTPOptions (IFTPOptions)		Read-only	An interface containing settings for sending the capture out to FTP (output set to **sioFTP** or **stoFTP**). At a minimum, server and filename must be set.
OutputMailOptions (IMailOptions)		Read-only	An interface for setting email output information. Either all information is specified and the email is sent immediately, or the user is prompted for all information.
OutputPrinterOptions (IPrinterOptions)		Read-only	An interface for use with printer output. If **UseDefault:Printer** is set to true then other values set programmatically by **SetData** or by the user with the **ShowSelectionDialog** are ignored.
TextFileOptions2 (ITextFile2)		Read-only	An interface containing settings for text file output with RTF capability. The file, regardless of the naming method used,

Property (type)	Default	Read / Write	Notes
			has a .txt extension. Only valid when using **stoFile** as the output mode for a text capture.
UseMagnifierWindow (Boolean)	True	Write	When enabled, a magnifier window shows next to the capture crosshairs for precision when defining capture regions.

ITextFile

The **ITextFile** interface contains all the settings for output text file (**stoFile**). The file, regardless of which file naming method is used, has a .txt extension. These options are only used when using **stoFile** as the output for a text capture object.

Properties

Property (type)	Default	Read / Write	Notes
AddInteractiveAnnotation (Boolean)	False	Write	Provides the user with a dialog to set the annotation text after the capture has finished.
AnnotationPrefix (string)	">>>"	Write	The string to add before the annotation. This string differentiates the captured text from the annotation text.
AppendToFile (Boolean)	False	Write	If true, the captured text is appended to the output file if it already exists. If it does not exist, the output file is created. If false, the Snagit COM server overwrites the existing file with the new capture data or creates a new file with the captured text added to it. If **FileNamingMethod** is set to **sofnmAuto**, then this property is ignored.
AutoNumPrefixDigits (long)	4	Write	The prefix of automatically named files generated by text captures when **FileNamingMethod** is set to **sofnmAuto**.
Directory (string)	"" (empty string)	Write	Sets the directory where the output file is created. Valid values are an existing directory on the system or an empty string. If **FileNamingMethod** is not set to **sofnmPrompt**, the Snagit COM server fails if this property is not set.
Filename (string)	"SNAGIT"	Write	Specifies the output filename. This property is only used if **FileNamingMethod** is set to **sofnmFixed**.
FileNamingMethod (snagOutputFileNamingMethod)	sofnmPrompt	Write	Specifies the method to use for naming the output text files.

ITextFile2

The **ITextFile2** interface contains all the settings for output text file. The file, regardless of which file naming method is used, has a .txt extension. These options are only used when using **stoFile** as the output for a text capture object.

Properties

Property (type)	Default	Read / Write	Notes
AddInteractiveAnnotation (Boolean)	False	Write	Provides the user with a dialog to set the annotation text after the capture has finished.
AnnotationPrefix (string)	">>>"	Write	The string to add before the annotation. This string differentiates the captured text from the annotation text.
AppendToFile (Boolean)	False	Write	If true, the captured text is appended to the output file if it already exists. If it does not exist, the output file is created. If false, the Snagit COM server overwrites the existing file with the new capture data or creates a new file with the captured text added to it. If **FileNamingMethod** is set to **sofnmAuto**, then this property is ignored.
AutoNumPrefixDigits (long)	4	Write	The prefix of automatically named files generated by text captures when **FileNamingMethod** is set to **sofnmAuto**.
Directory (string)	"" (empty string)	Write	Sets the directory where the output file is created. Valid values are an existing directory on the system or an empty string. If **FileNamingMethod** is not set to **sofnmPrompt**, the Snagit COM server fails if this property is not set.
Filename (string)	"SNAGIT"	Write	Specifies the output filename. This property is only used if **FileNamingMethod** is set to **sofnmFixed**.
FileNamingMethod (snagOutputFileNamingMethod)	sofnmPrompt	Write	Specifies the method to use for naming the output text files.
UseFormattedText (Boolean)	False	Write	If true, uses RTF formatted text. If false, uses plain text.

ITextFilters

The **ITextFilters** interface is a collection of filters that modify the text to output. Multiple filters may be used at the same time.

Properties

Property (type)	Notes
Annotation (**ITextAnnotation**)	An interface that adds a caption to a capture.
Font (**ISelectedArea**)	An interface that exposes the bounding box around the captured area. For a rectangular region, this is exactly the same. For an irregular region like a free-hand area, this is the bounding box around that area (leftmost point in the area, rightmost, etc).
Layout (**ITextLayout**)	An interface that changes the way the resulting text file is formatted.

ITextFont

The **ITextFont** interface specifies font properties for text output. Fonts also show in the Editor window.

Properties

Property (type)	Default	Read / Write	Notes
Height (long)	12	Write	Sets the height, in "ems," to use for the font. The system uses the nearest height to the specified value that does not exceed it.
Weight (long)	400	Write	Sets the weight, or thickness, of the font. Use a value of 400 for normal text and 700 for bold text. Valid values are 0 - 1000.
Italic (long)	0	Write	If true, the text is written in italics.
Underline (long)	0	Write	If true, the text is underlined.
PitchFamily (long)	1	Write	For more information, see the LOGFONT struct in the Microsoft documentation.
CharacterSet (long)	0	Write	For more information, see the LOGFONT struct in the Microsoft documentation.
FaceName (string)	"Arial"	Write	The font face name such as "Courier", "Times New Roman", or "Sans Serif" to use.

ITextLayout

The **ITextLayout** interface changes the way the resulting text file is formatted.

Properties

Property (type)	Default	Read / Write	Notes
CollapseBlankColumns (Boolean)	False	Write	Removes redundant blanks between text in a line. At one point, this property was spelled wrong- the 'n' and 'm' were reversed. As of Snagit 8.1.0, scripting languages (Automation/OLE/Dispatch languages) can use the correct spelling of **CollapseBlankColumns**.
Layout (**snagTextLayout**)	stlSpaceFormatted	Write	Defines how the text is laid out in the capture.
RemoveBlankLines (Boolean)	False	Write	If true, removes blank lines from the captured text.
TextDelimiter (string)	","	Write	Puts a delimiter character (comma, tab, semicolon, etc) between columns in the captured text. Using delimiters can be helpful if the capture is destined for a spreadsheet or database file. Use with **stlColumnDelmited**.
UseWordWrap (Boolean)	False	Write	Allows you to specify word wrap for the Editor window and for printing captured text.

ITWAINOptions

The **ITWAINOptions** interface contains all the settings for a TWAIN capture (input set to **siiTwain**).

Methods

Method	Action	Parameters	Notes
ShowSelectionDialog (Long, IParentHandle, Boolean dialogAccepted)	Displays a TWAIN device selection dialog to the user. If the user clicks OK in the dialog, the TWAIN device selected in the dialog becomes the selected TWAIN device for the Snagit capture object.	[in] **lParentHandle** (long): Handle to windows (hwnd) in 32-bit integer format. If handle is not available set value to 0. [out] **dialogAccepted** (Boolean): True if the user clicked OK to select a TWAIN device and false if the user clicked Cancel.	If this dialog is shown to the user and a TWAIN device is selected, it has no affect unless the input of the image capture object is set to **siiTwain**.

Properties

Property (type)	Default	Read / Write	Notes
Source (string)	"" (empty string)	Write	Specifies by name the TWAIN source to use.

IWindowOptions

The **IWindowOptions** interface contains the settings for use when capturing a window. These options are only valid when using **siiWindow** or **stiWindow** as the input type.

Properties

Property (type)	Default	Read / Write	Notes
SelectionMethod (snag-WindowSelectionMethod)	swsmInteractive	Write	Specifies the method used to select the window for capture.
Handle (string)	0 (long)	Write	Specifies the window handle of the window to capture.
XPos (string)	0 (long)	Write	Specifies the X coordinate of a point on the screen. Used in combination with the Y coordinate to capture the window under that point.
YPos (string)	0 (long)	Write	Specifies the Y coordinate of a point on the screen. Used in combination with the X coordinate to capture the window under that point.

Enumerations

The following section describes pre-defined data objects used throughout the Snagit COM server.

snagAutoScrollMethod

An enumeration used to scroll windows or regions automatically.

Name	Value	Description
sasmNone	0	Do not automatically scroll a window or region.
sasmVertical	1	Automatically scroll the window or region down vertically.
sasmHorizontal	2	Automatically scroll the window or region to the right horizontally.
sasmBoth	3	Automatically scroll the window or region both down and to the right.

snagAutoScrollStartingPosition

An enumeration used to set the starting location for automatically scrolling a window.

Name	Value	Description
sasspCurrent	0	Begin the automatic scroll from the current position.
sasspTop	1	Scroll to the top of the window before starting the automatic scrolling capture.
sasspLeft	2	Scroll to the left of the window before starting the automatic scrolling capture.
sasspTopLeft	3	Scroll to the top and left of the window before starting the automatic scrolling capture.

snagCaptionTextStyle

An enumeration used to set the text style in captions placed in images.

Name	Value	Description
sctsNormal	0	Text captions on an image have no special effect.
sctsDropShadow	1	Text captions on an image have a drop shadow.
sctsOutlinedShadow	2	Text captions on an image have an outlined shadow.

snagCaptureState

An enumeration used to set the state of the Snagit COM server during capture.

Name	Value	Description
scsIdle	0	The Snagit COM server is not currently taking a capture.
scsCaptureSucceeded	10	The capture was completed successfully.
scsCaptureFailed	11	The capture failed or was canceled.
scsBusy	12	The capture process has not yet completed (e.g. waiting for user input on a dialog).

snagColorConversionMethod

An enumeration used to change image colors.

Name	Value	Description
sccmNone	0	No color conversion.
sccmMonochrome	1	Change the captured image to monochrome (black and white).
sccmHalftone	2	Change the captured image to dithered monochrome.
sccmGrayscale	3	Change the captured image to grayscale.

snagColorSubMethod

An enumeration used to set how colors are replaced in an image.

Name	Value	Description
scsmNone	0	Leave the same colors in the image.
scsmInvert	1	Invert the color values (255 - x for each of Red, Blue, and Green).
scsmCustom	2	Use the values specified by the user.

snagCompassDirection

An enumeration used to set direction specifications.

Name	Value	Description
scdNorth	0	Up
scdNorthEast	1	Up and right
scdEast	2	Right
scdSouthEast	3	Down and right
scdSouth	4	Down
scdSouthWest	5	Down and left
scdWest	6	Left
scdNorthwest	7	Up and left

snagError

An enumeration used to set the type of error reported by the Snagit COM server.

Name	Value	Description
serrUnknown	-1	An unknown error occurred
serrNone	0	No error
serrSnagitExpired	1	Expired evaluation version of Snagit
serrInvalidInput	2	An invalid input was specified for the capture object
serrInvalidOutput	3	An invalid output was specified for the capture object
serrEngineBusy	4	The capture engine is currently busy
serrInvalidScrollDelay	5	The scroll delay specified for the automatic scrolling parameter was invalid
serrInvalidDelay	6	Delay capture has an invalid delay time
serrInvalidColorEffectValue	7	One of the values set for a color effect is invalid
serrInvalidFileProgressiveValue	8	The progressive value setting in **ImageFile** us not valid
serrInvalidFileQualityValue	9	The file quality setting in **ImageFile** is not valid
serrInvalidFileDirectory	10	A valid directory is not specified for **ImageFile** out-put
serrInvalidColorConversionValue	11	An invalid color conversion value was specified in the **ImageColorConversion** settings
serrInvalidImageResolution	12	An invalid image resolution value was specified in the **ImageResolution** settings

snagHotspotType

An enumeration used to set the types of controls to capture and create as hotspots.

Name	Value	Description
shtUnknown	-1	The hotspot type is unknown
shtNone	0	Do not capture hotspots
shtLinksOnly	1	Turn links found on webpages into hotspots in the image
shtLinksAndControls	2	Turn links, buttons, tabs, and other controls into hotspots in the image

snagImageColorDepth

An enumeration used to set the number of bits used to represent a color in an image.

Name	Value	Description
sicdAuto	0	Automatically choose the color depth for the image.
sicd1Bit	1	Use 1 bit to specify the color (monochrome)
sicd2Bit	2	Use 2 bit color
sicd3Bit	3	Use 3 bit color
sicd4Bit	4	Use 4 bit color
sicd5Bit	5	Use 5 bit color
sicd6Bit	6	Use 6 bis color
sicd7Bit	7	Use 7 bit color
sicd8Bit	8	Use 8 bit color
sicd16Bit	16	Use 16 bit color
sicd24Bit	24	Use 24 bit color (truecolor)
sicd32bit	32	Use 32 bit color (truecolor plus transparency information)

snagImageFileSubType

An enumeration used to set the subtypes for image file formats.

Name	Value	Description
sifstUnknown	-1	Unknown format
sifstBMP_Uncompressed	0x0000001	Uncompressed Windows bitmap
sifstBMP_RLE	0x0000002	Windows bitmap with Run-Length Encoding
sifstGIF_NonInterlaced	0x0000004	Non-interlaced GIF
sifstGIF_Interlaced	0x0000008	Interlaced GIF
sifstJFIF_444	0x0000010	JPEG non-progressive 4:4:4
sifstJFIF_422	0x0000020	JPEG non-progressive 4:2:2

Name	Value	Description
sifstJFIF_411	0x0000040	JPEG non-progressive 4:1:1
sifstJFIF_444_Progressive	0x0000080	JPEG progressive 4:4:4
sifstJFIF_422_Progressive	0x0000100	JPEG progressive 4:2:2
sifstJFIF_411_Progressive	0x0000200	JPEG progressive 4:1:1
sifstJFIF_Gray	0x0000400	JPEG non-progressive grayscale
sifstJFIF_Gray_Progressive	0x0000800	JPEG progressive grayscale
sifstCCITT	0x0001000	TIFF subtype for fax
sifstCCITT_Group3_1Dimension	0x0002000	TIFF subtype for fax
sifstCCITT_Group3_2Dimension	0x0004000	TIFF subtype for fax
sifstCCITT_Group4	0x0008000	TIFF subtype for fax
sifstTIF_Uncompressed	0x0010000	Uncompressed TIFF
sifstTIF_PackBits	0x0020000	PackBits compressed TIFF
sifstTIF_LZW	0x0040000	LZW compressed TIFF
sifstJTIF_Gray	0x0080000	JPEG compressed grayscale TIFF
sifstTIF_CMYK	0x0100000	Uncompressed TIFF using CMYK colorspace
sifstTIF_YCC	0x0200000	Uncompressed TIFF using YCC colorspace
sifstTIF_PACK_CMYK	0x0400000	PackBits compressed TIFF using CMYK colorspace
sifstTIF_PACK_YCC	0x0800000	PackBits compressed TIFF using YCC colorspace
sifstTIF_LZW_CMYK	0x1000000	LZW compressed TIFF using CMYK colorspace
sifstTIF_LZW_YCC	0x2000000	LZW compressed TIFF using YCC colorspace
sifstJTIF_444	0x4000000	JPEG compressed TIFF 4:4:4
sifstJTIF_422	0x8000000	JPEG compressed TIFF 4:2:2
sifstJTIF_411	0x10000000	JPEG compressed TIFF 4:1:1

snagImageFileType

An enumeration used to set the format of an image file.

> Formats with values from 7 to 27 are only available for Snagit version 8.1 and higher.

Name	Value	Description
siftUnknown	-1	Unknown format
siftBMP	0	Windows bitmap
siftTIFF	2	Tagged Image File Format
siftJPEG	3	JPEG
siftGIF	4	CompuServe GIF
siftPNG	5	Portable Network Graphic
siftTGA	6	Truevision Targa
siftRAS	7	SUN Raster
siftWMF	9	Windows Metafile
siftEPS	11	Encapsulated Postscript
siftOS2	17	OS/2 Bitmap
siftWFX	18	WinFax
siftEMF	19	Windows Enhanced Metafile
siftWPG	20	WordPerfect
siftPSD	21	Adobe PhotoShop 3.0
siftICO	22	Windows icon
siftCUR	23	Windows cursor
siftPDF	24	Adobe PDF
siftSNAG	25	Snagit Capture File
siftSWF	26	Adobe Shockwave Flash
siftMHT	27	Web Page with Image (MHTML)

snagImageInput

An enumeration used to set the type of capture.

Name	Value	Description
siiDesktop	0	Capture the full screen
siiWindow	1	Capture one window
siiRegion	4	Capture a region of the screen
siiGraphicFile	6	Capture an existing graphic file
siiClipboard	7	Capture the contents of the Clipboard
siiMenu	9	Capture a menu
siiObject	10	Capture an object (user interactive)
siiFreehand	12	Capture a user drawn freehand region of the screen
siiCustomScroll	18	User draws a region and then clicks a scroll button. Capture continues until the window has scrolled to the bottom.
siiTWAIN	19	Capture from a TWAIN source (usually a scanner)
siiExtendedWindow	23	REsize a window before capture. The size of the window (potentially larger than the screen resolution) is specified in advance or by the user.
siiCapture	25	Capture using the All-in-One Capture ® input, allowing window, region or scrolling window capture.

snagImageOutput

An enumeration used to set the output of a captured image.

Name	Value	Description
sioNone	0	No output is set
sioPrinter	1	Send the captured image directly to the printer
sioFile	2	Save the image to a file. The filename can be specified, automatically generated, or prompted for.
sioClipboard	4	Place the captured image in the system Clipboard

Name	Value	Description
sioMail	8	Send the image as an attachment in an email. This start the system default email client. Uses settings in **IMailOptions**.
sioFTP	32	Upload an image file to an FTP server. See **IFTPOptions**

snagImageScaleBy

An enumeration used to set the scaling dimensions of the image.

Name	Value	Description
sisbFactor	0	Scale the image by multiplicative factor
sisbWidthAndHeight	1	Scale to math a given width and height
sisbWidth	2	Scale width to match a specified value
sisbHeight	3	Scale height to match a specified value

snagImageScaleMethod

An enumeration used to change the size of a captured image.

Name	Value	Description
sismCustom	0	Use custom scaling options defined in IImageScale
sismPercentage	1	Scale by percentage
sismNone	2	Do not scale image

snagNotificationType

An enumeration used to set the types of messages Snagit displays. These are not related to event handling.

Name	Value	Description
sntNone	0	No notifications are displayed
sntInformation	1	Informational dialogs are displayed
sntErrors	2	Error messages are displayed
sntAll	3	Both error and informational dialogs are displayed

snagOutputFileNamingMethod

An enumeration used to determine how files are named during file output.

Name	Value	Description
sofnmPrompt	0	The user is prompted for the name and location to save the file. Use with **sioFile** and **stoFile**.
sofnmFixed	1	The file is saved with a fixed file name specified by the program. Specified in **IImageFile** or **IImageFile2**.
sofnmAuto	2	The file is saved with an automatically generated name.

snagPlacement

An enumeration used to set the placement of objects.

Name	Value	Description
spUnknown	-1	Placement unknown
spLeftTop	0	Place in upper left
spRightTop	1	Place in upper right
spLeftBottom	2	Place in lower left
spRightBottom	3	Place in lower right
spCenterMiddle	4	Place in the center
spCenterTop	5	Place in the center of the top
spLeftMiddle	6	Place in the middle of the left
spRightMiddle	7	Place in the middle of the right
spCenterBottom	8	Place in the center of the bottom
spOutsideTop	9	Place on the outside top of the capture
spOutsideLeft	10	Place on the outside left of the capture
spOutsideBottom	11	Place on the outside bottom of the capture
spOutsideRight	12	Place on the outside right of the capture

snagPrintScale

An enumeration used to set how an image is fitted on a page for printer output.

Name	Value	Description
spsSinglePage	0	The image, regardless of size, is scaled to fit on one page.
spsPercentScale	1	A scaling percentage is applied to decrease or increase the image's size onthe printed page. An image printed at 100 percent is the same as one printed with the **spsProportionalToScreen** setting.
spsProportionalToScreen	2	The relationship between the width of the area captured and the width of the screen is used to calculate the width of the printed image in relation to the width of the printed page.
spsFixedSize	3	The width and height are fixed, scaling or distorting the image if necessary.
spsSinglePageMaximize	4	The image is made as large as possible while still fitting on one page. This setting preserves the aspect ratio of the original image.
spsAutoScale	5	The width of the image is scaled to the width of the page. The length is scaled in proportion and can be more than one page.

snagRegionSelectionMethod

An enumeration used to specify what method to use when capturing a region.

Name	Value	Description
srsmInteractive	0	Interactively click and drag a region to be captured.
srsmFixed	1	Use values provided to capture a region of fixed size. This may still be interactive if a starting X and Y position is not given.

snagTextInput

An enumeration used to set the type of input for text captures.

Name	Value	Description
stiDesktop	0	Capture text from the entire screen
stiWindow	1	Capture text from one window
stiRegion	4	Capture text from within a specific region
stiClipBoard	7	Capture text from the Clipboard
stiObject	10	Capture text from an object. This could be a window, control, text box, static text in a dialog, etc. More general than the **stiWindow**

Name	Value	Description
		capture.
stiCustomScroll	18	User draws a region and then click a scroll button. Capture continues until the window has scrolled all the way to the bottom.

snagTextLayout

An enumeration used to filter text output.

Name	Value	Description
stlSpaceFormatted	0	Implies that the captured text is separated into groups by spaces
stlColumnDelimited	1	Implies that the captured text is aligned in columns

snagTextOutput

An enumeration used to set the order of time and date for timestamps.

Name	Value	Description
sioPrinter	1	Send the text directly to the printer
sioFile	2	Save the text in a TXT file
sioClipboard	4	Put the text on the Clipboard
sioMail	8	Attach the text in a TXT file and send it as an email
sioFTP	32	Upload a TXT file to an FTP server

snagTimeDateOrder

An enumeration used to set the order of time and date for timestamps.

Name	Value	Description
stdoTimeThenDate	0	Show time and then date
stdoTimeOnly	1	Show only the time
stdoDateThenTime	2	Show the date and then the time
stdoDateOnly	3	Show only the date

snagTrimMethod

An enumeration used to cut the outside borders of captured images.

Name	Value	Description
stmNone	0	No trimming is performed
stmManual	1	The edge of the image is trimmed by the number of pixels specified
stmAuto	2	Automatically trim blank areas from the edges of captured images. A blank area on an image is one that has no change in color.

snagWindowSelectionMethod

An enumeration used to specify how to make a window selection.

Name	Value	Description
swsmInteractive	0	The user selects a window to capture
swsmActive	1	Capture the currently active window
swsmHandle	2	Capture the window with the given window handle (as a long value)
swsmPoint	3	Capture the window under the point specified with X and Y coordinates

Events

The **_ICaptureEvents** interface is an event interface used for callbacks. A client program may choose to subscribe to events from the Snagit COM Object.

Methods

Name	Action	Parameters	DispID
OnError	Occurs when there is an error in the capture object.	[in] Error (**snagError**): The code for the error that occurred.	1
OnStateChange	Occurs when there is a state change in the capture object.	[in] State (**snagCaptureState**): The code for the state change that occurred.	2
OnFileWritten	Occurs when a file is written to notify the client program of the filename.	[in] Filename (string): The filename that was written.	3

Data Types

Boolean

True (-1) or False (0)

> Many languages have defines for true and false values; these may or may not be translated to the correct `VARIANT_BOOL` values. For example, C++/ATL programs should use the `VARIANT_TRUE` and `VARIANT_FALSE`, whereas Visual Basic, Visual Basic .NET, Visual Basic Script, and C# correctly translate the default `true` and `false` values.

Long

32-bit signed integer

String

Variable length string

- C++/ATL: `BSTR` or `CComBSTR`

Interface

All interfaces are derived from IDispatch for use in scripting languages such as Visual Basic Script

Colors

Colors are represented as long values where the format is :
`0x00bbggrr`

Languages such as MFC and Visual Basic define the macro `RGB(r, g, b)` that correctly creates this value from the three component values.

Code Samples

Example code for the Snagit COM server is available in the following languages:

- C++/ATL
- C#
- Visual Basic .NET
- Visual Basic Script
- Visual Basic

C++/ATL

Code samples for the C++/ATL language.

Import the table implementation file (Snagit.tlb)

```
#import "Path\To\Snagit.tlb" rename_namespace("Snagit")
```

Declare an object

- Snagit Version 6.2 and later

```
Snagit::IImageCapturePtr pImageCapture; // v6.2 and later
```

- Snagit Version 8.1 and later

```
Snagit::IImageCapture2Ptr pImageCapture; // v8.1 and later
```

Create an object

```
pImageCapture.CreateInstance( __uuidof( Snagit::ImageCapture ) );
```

Handle objects

To handle objects takes several steps. Capturing error events is illustrated here.

1. At the top of your class declaration file create an external reference to an _ATL_FUNC_INFO object.

   ```
   extern _ATL_FUNC_INFO SnagitErrorInfo;
   ```

2. Next have your class inherit from the IDispEventSimpleImpl templated class.

```
class CMyClass : IDispEventSimpleImpl<1, CMyClass, &__uui-
dof(Snagit::_ICaptureEvents)>
```

3. Next create a typedef inside your class declaration to talk about the events more easily.

```
typedef IDispEventSimpleImpl<1, CMyClass, &__uuidof(Snagit::_
ICaptureEvents)> SnagitErrorEvents;
```

4. Now create a sink map inside your class declaration to setup the message handling

```
BEGIN_SINK_MAP(CMyClass)

     SINK_ENTRY_INFO(1, __uuidof( Snagit::_ICaptureEvents
     ), 0x01, OnSnagitError, SnagitErrorInfo )

END_SINK_MAP()
```

5. Declare the message handler function inside your class declaration

```
void __stdcall OnSnagitError( Snagit::snagError nErrorCode );
```

6. Now in your class implementation file, define the SnagitErrorInfo _ATL_FUNC_INFO object

```
_ATL_FUNC_INFO SnagitErrorInfo = { CC_STDCALL, VT_EMPTY, 1, {
VT_I4 } };
```

7. Before you start getting events, you must advise the object that you are listening for events somewhere in the implementation file

```
SnagitErrorEvents::DispEventAdvise( pImageCapture );
```

8. Lastly, implement the handler function

```
void _stdcall CMyClass::OnSnagitError( Snagit::snagError nEr-
rorCode )

{

     /* implementation */

}
```

C#

Code samples for the C# language.

Declare an object

- Snagit Version 6.2 and later

```
private SNAGITLib.ImageCapture ImageCap; // v6.2 and later
```

- Snagit Version 8.1 and later

```
private SNAGITLib.IImageCapture2 ImageCap; // v8.1 and later
```

Create an object

```
ImageCap = new SNAGITLib.ImageCaptureClass();
```

Handle callbacks

To handle callbacks, create a function that handles the callback and attach it to the object using the correct delegate.

```
ImageCap.OnError += new SNAGITLib._ICaptureEvents_OnEr-
rorEventHandler( this.OnError );
```

Visual Basic .NET

Code sample for the Visual Basic .NET language.

Declare an object

- Snagit Version 6.2 and later

```
Public ImageCapture1 As SNAGITLib.ImageCapture ' v6.2 and later
```

- Snagit Version 8.1 and later

```
Public ImageCapture1 As SNAGITLib.IImageCapture2 ' v8.1 and later
```

Declare an object able to receive events

```
Public WithEvents ImageCapture1 As SNAGITLib.ImageCapture
```

Create an object

```
ImageCapture1 = New SNAGITLib.ImageCaptureClass()
```

If an object is created with the ability to receive events

```
'This function handles the OnStateChange event from the Image-
Capture object

Private Sub CapState(ByVal capState As SNAGITLib.snagCaptureState)
Handles ImageCapture1.OnStateChange

...

...

End Sub
```

Visual Basic Script

Code sample for the Visual Basic Script language.

Create an object

```
'Create and image capture object

set ImageCap = CreateObject("SNAGIT.ImageCapture")
```

If the script is allowed to exit before the capture has completed, the capture object will go out of scope and will exit. To keep the script running while the capture finishes a sleeping loop can be used.

```
Do Until ImageCap.IsCaptureDone

     WScript.Sleep 10

Loop
```

Visual Basic

Code sample for the Visual Basic language.

Declare an object

- Snagit Version 6.2 and later

```
Dim ImageCapture As SNAGITLib.ImageCapture ' v6.2 and later
```

- Snagit Version 8.1 and later

```
Dim ImageCapture As SNAGITLib.IImageCapture2 ' v8.1 and later
```

Declare an object able to receive events

```
Dim WithEvents ImageCapture As SNAGITLib.ImageCapture
```

Create an object

```
Set ImageCapture = CreateObject("Snagit.ImageCapture")
```

If an object is created with the ability to receive events

```
' This function handles the OnStateChange event fired by the Image-
Capture interface

' Event handling is done simply by putting the event after an _
after the object that

' fires the event. That is, Object_Event

Private Sub ImageCapture_OnStateChange(ByVal capState As SNA-
GITLib.snagCaptureState)
```