Wang Image Annotation Tool Button Control

The Wang **Image Annotation Tool Button** control adds image annotation functions to applications that support 32-bit OLE controls.   It enables end users to add text and graphical annotations to displayed image files.

Linking the Annotation Tool Button Control With the Image Edit Control

The **Image Edit** control is the main Imaging control.   The Image Annotation Tool Button control links to Image Edit control to permit the annotation of displayed images.   The Image Annotation Tool Button control sends messages to the Image Edit control that set annotation attributes, thereby changing all of the Image Edit annotation attributes to the values set within the Image Annotation Tool Button control.   When the **Draw** method is invoked, the Image Edit control draws the annotation.

Establishing the Link

To link the Image Annotation Tool Button control to the Image Edit control:

1. Set the **DestImageControl** property of the Image Annotation Tool Button control to the name of the desired Image Edit control.

2. Set the **ImageControl** property of the Image Edit control to the same Image Edit control name.

Annotation Types

The Image Annotation Tool Button Control supports the annotation types described in the following list:

Annotation Type	Description
Attach-a-Note	Enters text into a background rectangle on an image
Filled Rectangle	Covers a portion of an image
Freehand Line	Draws a freehand line on a section of text or a portion of an image for emphasis
Hollow Rectangle	Places a border around areas of an image for emphasis
Image Embedded	Embeds an actual copy of another image in an image file
Image Reference	Includes another image in an image file by reference (that is, it links to an external file that contains the image)
Select Annotations	Selects annotation marks for deleting, modifying, moving, or resizing
Straight Line	Underlines text, demarcates a section of a page, or draws callout lines Highlights text when drawn using the transparent line style
Text	Enters text directly on an image
Text From File	Enters text from a file on an image
Text Stamp	Places a text stamp directly on an image

See Also

Image Edit control

Imaging Concepts

Properties (Image Annotation Tool Button control)

AnnotationBackColor	AnnotationTextFile
AnnotationFillColor	AnnotationType
AnnotationFillStyle	DestImageControl
AnnotationFont	Enabled
AnnotationFontColor	hWnd
AnnotationImage	PictureDisabled
AnnotationLineColor	PictureDown
AnnotationLineStyle	PictureUp
AnnotationLineWidth	StatusCode
AnnotationStampText	Value

AnnotationBackColor Property

Returns or sets the background color of an Attach-a-Note annotation object.

Applies To

Image Annotation Tool Button control, Image Edit control

Syntax

*object.***AnnotationBackColor**[*=color*]

Data Type

Color

Remarks

Use the RGB format.   The default color is yellow (255, 255, 0).

To ensure that the text in an Attach-a-Note annotation object will be visible against the background, specify a background color that is significantly different from the font color set using the **AnnotationFontColor** property.

See Also

AnnotationFontColor property

Annotations

AnnotationType property

GetSelectedAnnotationBackColor method

SetSelectedAnnotationBackColor method

AnnotationFillColor Property

Returns or sets the color used to fill a Filled Rectangle annotation object.

Applies To

Image Annotation Tool Button control, Image Edit control

Syntax

*object.***AnnotationFillColor**[=*color*]

Data Type

Color

Remarks

Use the RGB format.   The default color is red (255, 0, 0).

See Also

AnnotationFontColor property

Annotations

AnnotationType property

GetSelectedAnnotationFillColor method

SetSelectedAnnotationFillColor method

AnnotationFillStyle Property

Returns or sets the pattern used to fill image and Filled Rectangle annotation objects.

Applies To

Image Annotation Tool Button control, Image Edit control

Syntax

*object.***AnnotationFillStyle**[*=value*]

Data Type

Integer (enumerated)

Setting	Description
0 (default)	Transparent -- The underlying image data is visible
1	Opaque -- The underlying image data is obscured

Remarks

The AnnotationFillStyle property is used with the following annotation types:

Filled Rectangle

Image Embedded

Image Reference

See Also

Annotations

AnnotationType property

GetSelectedAnnotationFillStyle method

SetSelectedAnnotationFillStyle method

AnnotationFont Property

Returns or sets a font object's properties.   Applies to all text annotation types.

Applies To

  Image Annotation Tool Button control, Image Edit control

Syntax

  *object.***AnnotationFont**

Data Type

  Font object

Remarks

  Use the AnnotationFont property of an object to identify the specific font object whose properties you want to use or set.   The following list shows the properties you can use or set, along with their data types, descriptions, and defaults:

Property	Data Type	Description	Default
Bold	Boolean	True -- Selects bold attribute False -- Deselects bold attribute	False
Italic	Boolean	True -- Selects italic attribute False -- Deselects italic attribute	False
Name	String	The desired font name	MS San Serif
Size	Single	The desired font size in points (0 to 2048)	12
Underline	Boolean	True -- Selects underline attribute False -- Deselects underline attribute	False
StrikeThrough	Boolean	True -- Selects strikethrough attribute False -- Deselects strikethrough attribute	False

  For example, the following code sets the properties of the font object identified by the AnnotationFont property of an Image Edit object:

```
Dim x As Font
Set x = Imgedit1.AnnotationFont
x.Bold = True
x.Italic = True
x.Name = "Arial"
x.SIZE = 16
x.Underline = True
x.StrikeThrough = True
```

  The AnnotationFont property is used with the following annotation types:

  Text

  Attach-a-Note

  Text From File

  Text Stamp

See Also

  Annotations

  AnnotationType property

  GetSelectedAnnotationFont method

  SetSelectedAnnotationFont method

AnnotationFontColor Property

Returns or sets the font color of text annotation objects.

Applies To

Image Annotation Tool Button control, Image Edit control

Syntax

*object.***AnnotationFontColor**[=*color*]

Data Type

Color

Remarks

Use the RGB format.   The default color is black (0, 0, 0).

The AnnotationFontColor property is used with the following annotation types:

Text

Attach-a-Note

Text From File

Text Stamp

See Also

AnnotationBackColor property

AnnotationFillColor property

Annotations

AnnotationType property

GetSelectedAnnotationFontColor method

SetSelectedAnnotationFontColor method

AnnotationImage Property

Returns or sets the fully-qualified filename of an image file.

Applies To

Image Annotation Tool Button control, Image Edit control

Syntax

*object.***AnnotationImage**[*=name*]

Data Type

String

Remarks

The name must be a fully-qualified filename.   DOS and Windows filenames are permissible.   File extensions are required as part of the filename.

The image file specified will be combined with the image currently displayed by the Image Edit control when the annotation is applied.

The AnnotationImage property is used with the following annotation types:

 Image Embedded

 Image Reference

See Also

Annotations

AnnotationType property

Display method

GetSelectedAnnotationImage method

Images

AnnotationLineColor Property

Returns or sets the line color for line and Hollow Rectangle annotation objects.

Applies To

Image Annotation Tool Button control, Image Edit control

Syntax

*object.***AnnotationLineColor**[*=color*]

Data Type

Color

Remarks

Use the RGB format.   The default color is red (255, 0, 0).

The AnnotationLineColor property is used with the following annotation types:

Freehand Line

Hollow Rectangle

Straight Line

See Also

Annotations
AnnotationType property
GetSelectedAnnotationLineColor method
SetSelectedAnnotationLineColor method

AnnotationLineStyle Property

Returns or sets the line style for line and Hollow Rectangle annotation objects.

Applies To

Image Annotation Tool Button control, Image Edit control

Syntax

*object.***AnnotationLineStyle**[*=value*]

Data Type

Integer (enumerated)

Setting	Description
0 (default)	Transparent -- The underlying image data is visible
1	Opaque -- The underlying image data is obscured

Remarks

The AnnotationLineStyle property is used with the following annotation types:

Freehand Line

Hollow Rectangle

Straight Line

See Also

AnnotationLineWidth property

Annotations

AnnotationType property

GetSelectedAnnotationLineStyle method

SetSelectedAnnotationLineStyle method

AnnotationLineWidth Property

Returns or sets the line width for line and Hollow Rectangle annotation objects.

Applies To

Image Annotation Tool Button control, Image Edit control

Syntax

*object.***AnnotationLineWidth**[*=value*]

Data Type

Integer

Remarks

The integer value specifies the line width in pixels.   The valid range is from 1 to 999 pixels; the default value is 2 pixels.

The AnnotationLineWidth property is used with the following annotation types:

Freehand Line

Hollow Rectangle

Straight Line

See Also

AnnotationLineStyle property

Annotations

AnnotationType property

GetSelectedAnnotationLineWidth method

SetSelectedAnnotationLineWidth method

AnnotationStampText Property

Returns or sets the format of the stamp text to be placed on an image by a Text Stamp annotation. The stamp text can consist of text, text macro(s), or a combination of both.

Applies To

Image Annotation Tool Button control, Image Edit control

Syntax

*object.***AnnotationStampText**[=*stamp*]

Data Type

String

Remarks

Text is placed on the image when one of the following events occurs:
- The end user releases the left mouse button on a displayed image
- The **Draw** method is invoked

The stamp text can include the following text macros:

Macro	Description
%b	Alphabetic abbreviation of the month; for example, Jan
%B	Alphabetic full name of the month; for example, January
%d	ASCII numeric day of the month, ranging from 1 to 31
%H	ASCII time in a 24-hour, hh:mm format; where: hh = hour mm = minute
%I	ASCII time in a 12-hour, hh:mm Xm format; where: hh = hour mm = minute Xm = AM or PM
%m	ASCII numeric designation of the month, ranging from 1 to 12 for January through December respectively
%x	Default Windows date format
%X	Default Windows time format
%y	ASCII numeric last two digits of the year; for example, 95 for 1995
%Y	ASCII numeric full year; for example, 1995

Examples

- To create an annotation text stamp that displays the current time and date in a 12-hour format, enter:

        Time: %I Date: %d-%B-%y

    The resulting annotation text stamp is:

        Time: 10:38 pm Date: 07-November-95

▪ To create an annotation text stamp that displays the current time and date in a 24-hour format, enter:

Time: %H Date: %d-%B-%y

The resulting annotation text stamp is:

Time: 22:38 Date: 07-November-95

See Also

Annotations

AnnotationType property

Draw method

AnnotationTextFile Property

Returns or sets the fully-qualified filename of a text file to embed into an image.

Applies To

Image Annotation Tool Button control, Image Edit control

Syntax

*object.***AnnotationTextFile**[=*filename*]

Data Type

String

Remarks

The name must be a fully-qualified filename.   DOS and Windows filenames are permissible.   A file extension is required for all files.

The AnnotationTextFile property is used with the Text From File annotation type.

See Also

Annotations

AnnotationType property

AnnotationType Property

Returns or sets the type of annotation to be drawn.

Applies To

Image Annotation Tool Button control, Image Edit control

Syntax

*object.***AnnotationType**[=*value*]

Data Type

Integer (enumerated)

Setting	Description
0 (default)	None
1	Straight Line
2	Freehand Line
3	Hollow Rectangle
4	Filled Rectangle
5	Image Embedded
6	Image Reference
7	Text
8	Text Stamp
9	Text From File
10	Attach-a-Note
11	Select Annotations

Prerequisite Properties

Some annotation types require that other properties be set to valid values before they can be used in an annotation.   The following list shows the properties that must be set before the corresponding annotation type can be drawn:

Prerequisite Property	Annotation Type
AnnotationImage	Image Embedded
AnnotationImage	Image Reference
AnnotationStampText	Text Stamp
AnnotationTextFile	Text From File

The **Draw** method generates an error if it is invoked without the required property being set.

If the end user attempts to draw an annotation using the Image Annotation Tool Button control without the required property being set, an error event is sent to the target Image Edit control window.   It is up to you to decide whether to present this error to the end user in your application.

When using the standard Annotation Tool Palette, a Select File dialog box appears at run-time to allow the end user to select the desired file.

Do not attempt to change the AnnotationType property when the standard Annotation Tool Palette is displayed.   Use the **SelectTool** method instead.

Remarks

The Image Embedded annotation type (setting 5) stores a copy of the actual image data.   The

Image Reference annotation type (setting 6) stores the image data by reference (filename only).

See Also

AnnotationImage property

Annotations

AnnotationStampText property

AnnotationTextFile property

Draw method

SelectTool method (Image Edit control)

ShowAnnotationToolPalette method (Image Edit control)

DestImageControl Property

Returns or sets the link to the Image Edit control.

Applies To

Image Annotation Tool Button control

Syntax

*object.***DestImageControl**[=*controlname*]

Data Type

String

Remarks

Set the DestImageControl property to the name of the desired Image Edit control.   The **ImageControl** property of the Image Edit control must also be set to the same name.

The **Image Annotation Tool Button** control links to Image Edit control to permit the annotation of displayed images.   The Image Annotation Tool Button control sends messages to the Image Edit control that set annotation attributes, thereby changing all of the Image Edit annotation attributes to the values set within the Image Annotation Tool Button control.   When the **Draw** method is invoked, the Image Edit control draws the annotation.

See Also

Draw method

ImageControl property

Enabled Property

Enables or disables the control.

Applies To

Image Annotation Tool Button control, Image Edit control, Image Thumbnail control

Syntax

*object.***Enabled**[={*True|False*}]

Data Type

Boolean

Setting	Description
True (default)	Control is enabled (can respond to events)
False	Control is disabled (cannot respond to events)

hWnd Property

Indicates the window handle to the control.

Applies To

Image Annotation Tool Button control, Image Edit control, Image Thumbnail control

Syntax

[*object.*]**hWnd**

Data Type

Long

Remarks

Available at run-time as read-only.

PictureDisabled Property

Returns or sets the bitmap image that appears on the control when its **Enabled** property is set to False (disabled).

Applies To

Image Annotation Tool Button control

Syntax

*object.***PictureDisabled**[=*picture*]

Data Type

Picture

Remarks

The bitmap is scaled to fit the size of the control.   If no bitmap is specified, the default bitmap is used.

See Also

Enabled property

PictureDown property

PictureUp property

PictureDown Property

Returns or sets the bitmap image that appears on the control when it is selected.   A control is selected when an end user clicks on it, or when its **Value** property is set to True within a program.

Applies To

Image Annotation Tool Button control

Syntax

*object.***PictureDown**[*=picture*]

Data Type

Picture

Remarks

The bitmap is scaled to fit the size of the control.   If no bitmap is specified, the default bitmap is used.

See Also

PictureDisabled property

PictureUp property

Value property

PictureUp Property

Returns or sets the bitmap image that appears on the control when it is enabled but not selected.   A control is enabled when its **Enabled** property is set to True.   A control is not selected (deselected) when the end user has not clicked on it, or when its **Value** property is set to False within a program.

Applies To

Image Annotation Tool Button control

Syntax

*object.***PictureUp**[*=picture*]

Data Type

Picture

Remarks

The bitmap is scaled to fit the size of the control.   If no bitmap is specified, the default bitmap is used.

See Also

Enabled property

PictureDown property

PictureDisabled property

Value property

Value Property

Selects or deselects the control at run-time.   Setting this property to True selects the control, which is the same as an end user clicking on it.   Setting this property to False deselects the control.

When a control is selected, the following events occur:
- The bitmap specified in the **PictureDown** property is drawn on the control
- Other controls on the form with the same **DestImageControl** property value have their Value property set to False automatically
- The **AnnotationType** property in the Image Edit control is changed to the value of the AnnotationType property in the Image Annotation Tool Button control.

When a control is deselected, the bitmap specified in the **PictureUp** property is drawn on the control.

Applies To

Image Annotation Tool Button control

Syntax

*object.***Value**[=*{True|False}*]

Data Type

Boolean

Setting	Description
True	Control is selected
False (default)	Control is deselected

Remarks

This property is available only at run-time.

See Also

AnnotationType property

DestImageControl property

PictureDown property

PictureUp property

Methods (Image Annotation Tool Button control)

AboutBox

Draw

Draw Method

Draws the annotation.

Applies To

Image Annotation Tool Button control, Image Edit control

Syntax

*object.***Draw** *Left,Top*[*,Width,Height*]

Arguments

Parameter	Data Type	Setting
Left	Long	The starting horizontal coordinate of the annotation in image pixels
Top	Long	The starting vertical coordinate of the annotation in image pixels
Width	Long	The width of the annotation in image pixels
Height	Long	The height of the annotation in image pixels

Return Value

None

Remarks

The Draw method draws the annotation specified by the **AnnotationType** property, standard Annotation Tool Palette, or **SelectTool** method on the displayed image.   The method draws the annotation in accordance with the coordinates specified in the left, top, width, and height parameters.

The width and height parameters must be specified when using the following annotation types:

Select Annotations (optional; see the following Note)

Freehand Line

Filled Rectangle

Hollow Rectangle

Straight Line

Attach-a-Note

Note

Width and height parameter entry is optional when using the Select Annotations annotation type.   If only the left and top parameters are specified, selection can be made by point.   If all of the parameters are specified, selection can be made using the Select Annotations rectangle.

See Also

Annotations

Events (Image Annotation Tool Button control)

The following Events, used by the Image Annotation Tool Button control, are all standard:

Click	KeyUp
DblClick	MouseDown
Error	MouseMove
KeyDown	MouseUp
KeyPress	

Visual Basic adds the **DragDrop**, **DragOver**, **GotFocus**, and **LostFocus** events.

RGB Format

RGB (Red Green Blue) values range from 0 to 16,777,215 (&HFFFFFF).   The high byte of a number in this range equals 0; the lower 3 bytes, from least to most significant byte, determine the amount of red, green, and blue, respectively.   The red, green, and blue components are each represented by a number between 0 and 255 (&HFF).

Imaging Concepts

The following terms describe the imaging concepts that are associated with Imaging OLE controls:

Images

[What Is an Image?](#)

[Bitmap Image](#)

[Black-and-White Image](#)

[Color Image](#)

[File Types](#)

[Gray-Scale Image](#)

[Image Files](#)

[Image Page](#)

[Image Types](#)

[True Color Image](#)

[Units of Measure](#)

Annotations

[What Are Annotations?](#)

[Annotation Functions](#)

[Annotation Groups](#)

[Annotation Tool Palette](#)

[Annotation Types](#)

Compression

[What Is Compression?](#)

[Compressed Image](#)

[Compression Options](#)

[Compression Type Support](#)

[Compression Types](#)

[Uncompressed Image](#)

Display

[What Are Display Functions?](#)

[Display Types](#)

Palettes

[What Is a Palette?](#)

[Common Palette](#)

[Custom Palette](#)

Scanning

[What Is Scanning?](#)

[TWAIN Specification](#)

Units of Measure

Annotations

The following terms describe the annotation concepts that are associated with Imaging OLE controls:

What Are Annotations?
Annotation Functions
Annotation Groups
Annotation Tool Palette
Annotation Types

Compression

The following terms describe the compression concepts that are associated with Imaging OLE controls:

What Is Compression?

Compressed Image

Compression Options

Compression Type Support

Compression Types

Uncompressed Image

Display

The following terms describe the display concepts that are associated with Imaging OLE controls:

What Are Display Functions?
Display Types

Images

The following terms describe the image concepts that are associated with Imaging OLE controls:

Palettes

The following terms describe the palette concepts that are associated with Imaging OLE controls:

What Is a Palette?

Common Palette

Custom Palette

Scanning

The following terms describe the scanning concepts that are associated with Imaging OLE controls:

What Is Scanning?

TWAIN Specification

Annotations

The following terms describe the annotation concepts that are associated with Imaging OLE controls:

What Are Annotations?

Annotation Functions

Annotation Groups

Annotation Tool Palette

Annotation Types

Compression

The following terms describe the compression concepts that are associated with Imaging OLE controls:

What Is Compression?

Compressed Image

Compression Options

Compression Type Support

Compression Types

Uncompressed Image

Display

The following terms describe the display concepts that are associated with Imaging OLE controls:

Images

The following terms describe the image concepts that are associated with Imaging OLE controls:

Palettes

The following terms describe the palette concepts that are associated with Imaging OLE controls:

Scanning

The following terms describe the scanning concepts that are associated with Imaging OLE controls:

What Is Scanning?

TWAIN Specification

Annotation Functions

Annotation functions provide tools for drawing and manipulating annotation marks.   Depending on how you implement them in your application, the end user can:
- Create, move, resize, edit, and delete annotation marks.
- Modify annotation attributes (such as color, size, text, and visibility).
- Group annotation marks and control their visibility.
- Save annotation marks as part of the image or separately.

Annotation Groups

Annotation groups make it possible to organize and manage annotations, giving the end user the capability to:

- Logically organize annotations into functional groups; for example, Accounts Payable Group, Accounts Receivable Group, Purchasing Group, Receiving Group, and so on.
- Toggle the visibility of the groups; for example, to Turn On Accounts Payable Comments, Turn On All Comments, and so on.

A group can contain one or more annotation marks and multiple groups can be associated with a single image.

Annotation Types

Imaging OLE controls support the following annotation types:

Annotation Type	Description
Attach-a-Note	Enters text into a background rectangle on an image
Filled Rectangle	Covers a portion of an image
Freehand Line	Draws a freehand line on a section of text or a portion of an image for emphasis
Highlighter	Draws a transparent, filled rectangle for highlighting emphasis (available as a discreet annotation type on the Annotation Tool Palette)
Hollow Rectangle	Places a border around areas of an image for emphasis
Image Embedded	Embeds an actual copy of another image in an image file
Image Reference	Includes another image in an image file by reference (that is, it links to an external file that contains the image)
Rubber Stamp	Places an image or text stamp directly on an image (available as a discreet annotation type on the Annotation Tool Palette)
Select Annotations	Selects annotation marks for deleting, modifying, moving, or resizing
Straight Line	Underlines text, demarcates a section of a page, or draws callout lines
Text	Places text directly on an image
Text From File	Places text from a file on an image
Text Stamp	Places a text stamp directly on an image

What Are Annotations?

Annotations are digitized versions of the marks commonly used on documents in a paper-based environment; for example, highlight, rubber stamps, and text notes.   Like their paper-based counterparts, digitized annotation marks are typically used to prepare documents for subsequent handling in a business process.   However, digitized annotations provide additional benefits that paper-based tools do not:

- Digitized annotations can be added, moved, and deleted at will.
- The attributes of digitized annotations (such as color, size, text, and visibility) can be readily modified.
- Groups of digitized annotations can be named.   The named groups can then be selectively acted upon.

Annotation marks can be saved within a TIFF image file as annotation data that is kept separate from the image data.   The marks can also be merged with the image data in a process known as burning-in.   To save annotations to any file type other than TIFF, the annotations must be burned-in.

Burning-in converts the annotation marks into image pixels and merges them with the underlying image pixels in the process.   Once annotations are burned-in, they become part of the base image and can no longer be manipulated by the annotation functions.

Black-and-White Image

A Black-and-White image is a type of image where each pixel is either on (black) or off (white).   This type of image is typically used in fax transmissions.

Bitmap Image

A Bitmap image is a representation of a graphic image, displayed by setting individual pixels to white, black, or a color.   This type of image is typically used by the Windows Paintbrush applet and has a filename extension of .BMP.

Color Image

A Color image is a type of image that contains various colors.   Imaging OLE controls support 4- and 8-bit palettized images, as well as 24-bit color images.

Compressed Image

A Compressed image is an image that is stored using a technique that reduces the size of the image, thereby saving storage space.

What Is Compression?

Compression is the process by which image data is reduced to require less storage space. Employing compression is particularly important when saving gray-scale or color images because of their large size.

Compression Options

Compression options are bitwise values that dictate how image data is compressed.   A relationship exists between compression options and compression types, where certain compression options apply only to specific compression types.

Whenever a compression option and a compression type are specified as arguments to an Imaging OLE method, the software checks the relationship for validity.

Compression Option Support

Compression Option Support

Imaging OLE controls support the following compression options:

Compressed LTR -- Compressed bit order, Left-To-Right.   The compressed data codes begin at the most significant bit of each byte and are ordered from the most significant bit to the least significant bit.   The Compressed LTR compression option applies to all compression types that support black-and-white images, except the PackBits compression type.

EOL -- End Of Line.   Each line of data is terminated with a standard end-of-line bit.   The EOL compression option applies to the Group3(1D) compression type.

Expand LTR -- Expanded bit order, Left-To-Right.   The expanded data codes begin at the most significant bit of each byte and are ordered from the most significant bit to the least significant bit. The Expand LTR compression option applies only to image data that is being read; it is therefore meaningless when writing or converting images.   The Expand LTR compression option applies to all compression types except JPEG.

Negate -- A high-level request to invert black-and-white data.   The Negate compression option applies to all compression types that support black-and-white images.

Packed Lines -- The compressed data is packed into bytes with no padding at the end of each line (the data is not byte-aligned).   The Packed Lines compression option applies to the Group3(1D) and Group4(2D) compression types.

Prefixed EOL -- Each strip of data is prefixed by a standard end-of-line bit sequence.   The Prefixed EOL compression option requires that the EOL compression option be set as well.   It applies to the Group3(1D) compression type.

JPEG Options -- Joint Photographics Experts Group compression options.   JPEG compression options comprise all combinations of high, medium, or low compression and high, medium, or low quality (resolution).

The higher the compression, the lower the disk space requirement; the higher the resolution, the greater the image quality.   An image compressed with the Low Compression/High Quality option has the highest disk space requirement and the highest image quality.   Conversely, an image compressed with the High Compression/Low Quality option has the lowest disk space requirement and the lowest image quality.

The JPEG compression options apply to the JPEG compression type, which can be used with JPG and TIFF file types.

Compression Type Support

Imaging OLE controls support the following compression types:

File Type	Image Type	Compression Type Supported
AWD	Black-and-White	RBA (Microsoft proprietary)
BMP	Black-and-White Palettized4 Palettized8 BGR24	No compression
DCX/PCX (read-only)	Black-and-White Palettized4 Palettized8 RGB24	PCX proprietary
JPG (read-only)	Gray8 RGB24	JPEG
TIFF	Black-and-White Gray4 Gray8 Palettized4 (read-only) Palettized8 RGB24	No compression, Group3(1D), Group3(MH), Group4(2D), PackBits No compression No compression, JPEG No compression No compression No compression, JPEG

Compression Types

Imaging OLE controls support the following compression types:

RBA -- The RBA compression type is used to compress black-and-white AWD (At Work Document) images.   RBA is a Microsoft-proprietary compression type.

Group3(1D) -- A compression type based on the CCITT Group3 fax standard, Group3(1D) is used to compress black-and-white TIFF image files.   Group 3(1D) encodings are designed to provide reliable delivery of image data over inherently unreliable data links.   It is commonly used to compress images for fax transmission.

Group3(Modified Huffman [MH]) -- Similar to the Group3(1D) compression type, except Group3(MH) provides no end-of-line characters.   It is used to compress black-and-white TIFF image files.

Group4(2D) -- Based on the CCITT Group4 fax standard, Group4(2D) is used to compress black-and-white TIFF image files. Group 4(2D) encodings are designed to provide delivery of image data over reliable data links, such as ISDN or X.25.   It provides superior compression for non-dithered images when compared to the Group3(1D) compression type.

JPEG -- The Joint Photographics Experts Group compression type is used to compress 24-bit color (RGB24) and 8-bit gray-scale (Gray8) image files.   JPEG produces a high compression ratio, but some image data is actually lost during compression.

PackBits -- A simple, byte-oriented, run-length compression type, PackBits is used to compress black-and-white TIFF image files.

PCX -- A ZSoft-proprietary compression type, PCX is used to compress multipage DCX and single-page PCX black-and-while and color images.

Common Palette

A common palette is shared by all images being displayed. It is a special, 8-bit palette that has a variety of colors, chosen because they give generally good results for both color and gray-scale images. While the resulting image display is not as good as a custom palette rendition of the image, the common palette has the advantage of permitting any number of images to be displayed simultaneously with good overall quality.

Custom Palette

A custom palette is optimized for the display of a particular image.   An image displayed using a custom palette results in an image that is most like its original.

Using the custom palette has one disadvantage, which is only noticeable when displaying multiple images using a monitor and controller that display 256 or fewer colors.   The disadvantage is that background images can appear in lower quality than their originals.   This occurs because Windows shares the active palette of the foreground window, which may not be suitable for displaying background images.

What Are Display Functions?

Display functions provide the ability to show and manipulate images within an image window.
Typical display functions include:
- Copying, cutting, and pasting to the Clipboard
- Flipping
- Rotating
- Scrolling
- Zooming

These display functions manipulate the image data in the display memory of the PC.   The image file data is not modified unless the image is saved.

Display Types

The display type pertains to the way in which data is presented to the monitor.   It does not alter the original image in any way.   Imaging OLE controls support the following display types:

Black-and-White (Bi-Level) -- Displays an image as black and white.   The Black-and-White display type is typically used for applications that manage black-and-white documents exclusively.

Common -- Displays an image using the common palette.   The Common display type permits the display of multiple images simultaneously while maintaining reasonable image quality.   A monitor and controller capable of displaying 256 colors are recommended.

Custom -- Displays an image using the best palette possible.   The palette used depends on the type of image being displayed, as described in the following list:

Image	Palette Used
Black-and-White	Binary
4-Bit Gray-Scale	Common
8-Bit Gray-Scale	Gray8
24-Bit Color Images	Common

Because the palette used is designed specifically for the type of image being displayed, the custom display type is best suited to application programs that display one image at a time.   A monitor and controller capable of displaying 256 colors are recommended.

Gray8 -- Displays an image using a linear-scale palette of up to 128 shades of gray.

RGB24 -- Displays an image by letting Windows map the image directly to the monitor.   With suitable hardware, the RGB24 display type produces the best color at the optimum speed.   Further, it allows multiple images to be displayed with no loss in background image quality.   A monitor and controller capable of displaying 32,000 colors are recommended.

Dots Per Inch

Dots Per Inch (dpi) indicates the number of dots (or pixels) per inch in an image.   The more dots per inch in an image, the higher its density and quality.

File Types

Imaging OLE controls support the following file types:

AWD -- The file format for Microsoft At Work Document (AWD) images, AWD supports black-and-white images only.   AWD files are compressed using RBA compression.

BMP -- The file format used by the Windows Paintbrush applet.   BMP (bitmap) supports single-page color images.   Compression of BMP files is not supported.

DCX (read-only) -- The file format for multipage ZSoft PaintBrush images.   DCX supports both black-and-white and color images.   DCX files are compressed using PCX compression.

JPG (read-only) -- The file format for Joint Photographics Experts Group (JPEG) images.   JPG supports single-page 8-bit gray-scale and RGB24 color images.   JPG files are compressed using JPEG compression.

PCX (read-only) -- The file format for single-page ZSoft PaintBrush images.   PCX supports both black-and-white and color images.   PCX files are compressed using PCX compression.

TIFF -- The file format for Tag Image File Format (TIFF) images.   TIFF supports multipage black-and-white, gray-scale, and color images; several compression types; and a variety of options.   TIFF is compatible with all image densities.

Gray-Scale Image

A Gray-Scale image is a type of image that contains various shades of gray.   Imaging OLE controls support 4- and 8-bit gray-scale images.

What Is an Image?

An image is a photograph, line drawing, or text document that has been converted to digitized form, usually by a scanning device.   A scanned image is saved in a particular image file format.

Typically, an image is displayed in a window that can be sized or moved.   Images can also be annotated, e-mailed, faxed, or printed.

Image Files

Images are stored in image files, which are standard Windows or DOS files.   Image files can be organized in folders (directories), copied, renamed, deleted, and so on, using standard Windows or DOS commands.

Depending on the file format, image files can contain one or more images.   In multipage image files, each image is stored in a single image page.   Image files can be stored on the hard drive of a local PC, or on redirected network drives.   Applications can access any available image file by specifying its proper path (drive, directory, and filename).

Image Page

Image pages provide a way to organize multiple images in an image file.   Each page contains or stores one image.   The following file types support multiple pages:

- AWD
- DCX
- TIFF

Image Types

Imaging OLE controls support the following image types:

BGR24 -- This type of image uses 24 bits of data per pixel (eight bits-per-pixel for each of the three primary colors -- Blue, Green, and Red).   This translates to 256 shades of each primary color to provide up to 16 million colors for each image.   To display BGR24 images, the monitor and controller must be capable of displaying at least 32,000 colors.   The BGR24 image type is typically encountered in BMP files.

Black-and-White (Bi-Level) -- The Black-and-White image type displays one bit of data per pixel on the screen as either black or white.

Gray Scale -- There are two types of gray-scale images: Gray4 and Gray8.

Gray4 -- A Gray4 image uses 4 bits of data per pixel.   Every pixel on the screen can be displayed in 16 shades of gray, giving the image much better clarity than black-and-white images.

Gray8 -- A Gray8 image uses 8 bits of data per pixel.   Every pixel can be displayed with 256 shades of gray, producing better image clarity than Gray4.   Gray8 images are particularly effective for storing pictures and photographs.

Palettized -- There are two types of palettized images: Palettized4 and Palettized8.

Palettized4 -- A Palettized4 image uses 4 bits of data per pixel to reference a palette of 16 RGB (Red, Green, Blue) colors. These images can use a custom palette only.

Palettized8 -- A Palettized8 image uses 8 bits of data per pixel to reference a palette of 256 RGB colors.   These images can use either a custom or common palette.

RGB24 -- Also known as true color, an RGB24 image is the same as a BGR24 image, but with the colors (Red, Green, and Blue) in a different order.

What Is a Palette?

A palette is a look-up table that is used to translate an index value into a color value (known as an RGB value, for Red, Green, Blue).   There are two types: Common and Custom.

Color images are often stored using a palette so that they take up less space.   Such an image is said to be palettized.   A complex image that has been palettized is often of lower quality than the original image.   This is due to the fact that the eye can distinguish many more colors than are usually available in the palette.

Windows reserves 20 palette entries to use for window backgrounds, borders, and so on.

Pixel

The pixel is the smallest graphic unit on a screen or image.   It is equal to one screen dot or picture element.

Resolution

Resolution is the overall measurement of image quality.   It is expressed as the number of dots- or pixels-per-inch.   It is possible for an image to have different horizontal and vertical resolutions.   For example, images received from fax transmissions typically have resolutions of 200 by 100 pixels-per-inch.   The more pixels-per-inch, the higher the resolution.   Image quality increases in proportion with the resolution value.   Therefore, the higher the resolution, the greater the image quality.

What Is Scanning?

Scanning is the procedure for capturing data as images in electronic form.

An image acquisition device (typically a scanner) is connected to the PC and controlled by an application program.   The application program communicates with the scanner through an interface called the scanner data source (driver), which is usually supplied by the manufacturer of the scanner. Imaging OLE controls support TWAIN, the industry-standard specification for image capture.

True Color Image

A True Color image is a type of image that mixes shades of red, green, and blue to provide over 16 million colors.   These color images are also known as BGR24 or RGB24 images.

TWAIN Specification

TWAIN is an industry-standard, open specification that defines the way in which an image acquisition device (typically a scanner) communicates with an application program.   With TWAIN, an application program can acquire images through any device that complies with its specification.

For your scanner, or other image acquisition device, to function properly, you need the TWAIN data source from its manufacturer as well as the 32-bit TWAIN DLLs (Version 1.6).

Uncompressed Image

An Uncompressed image is an image that is stored as is, without compression of any kind.   This type of image is usually larger than a compressed image; it may therefore require a greater amount of storage space.

Out of Memory

More memory was required than is available.
- There are too many applications, documents, or source files open.

Close any unnecessary applications, documents, or source files that are open.
- There is a module or procedure that is too large.

Break extremely large modules or procedures into smaller ones.
- There are terminate-and-stay resident programs running.

Eliminate terminate-and-stay resident programs.
- There are too many device drivers loaded.

Eliminate unnecessary device drivers.

File Not Found

This file cannot be found.
- A statement refers to a file that does not exist.

Check the spelling of the filename, and the path specification.

File Already Exists

You tried to create a file with the same name as an existing file.
- Specify a new name, or delete the old file.

Disk Full

There is not enough room on the disk to complete the requested operation.
- Move some files to another disk, or delete some files on the current disk.
- Check your Windows\Temp directory or your Recycle Bin for files that you can safely delete.

Bad Filename

An error occurred trying to access the specified file.
- There is an invalid filename.

Filenames must conform to operating system conventions, as well as those of the programming language.   Filenames can be up to 255 characters.   Respecify the filename, and then retry your operation.

Path File Access Error

During a file or disk-access operation, the operating system could not make a connection between the path and the filename.

▪ The file specification is not correctly formatted.

A filename can contain a fully-qualified (absolute) or relative path.   A fully-qualified path starts with the drive name (if the path is on another drive) and then lists the explicit path from the root to the file. Any path that is not fully-qualified is relative to the current drive and directory.

▪ You attempted to save a file that would replace an existing read-only file.

From Windows Explorer, right-mouse click on the file, and select Properties from the pop-up menu. From the Properties window, change the properties to enable read/write capability.

Path Not Found

The path to a file comprises the drive specification plus the directories and subdirectories used to locate the file.   A path can be relative or absolute.

- During the file or disk-access operation, the operating system was unable to find the specified path.   Check the pathname and its syntax and retry your operation.

Invalid File Format

The file format you have specified is invalid.   The supported file types include:
- Tag Image File Format (TIF):   The OCX supports read/write access.
- At Work Document (AWD):   The OCX supports read/write access.
- PC Paintbrush Format (PCX):   The OCX supports read access.
- Multi-Page Paintbrush Format (DCX):   The OCX supports read access.
- Bitmap Image (BMP):   The OCX supports read/write access.
- Joint Photographic Experts Group (JPG):   The OCX supports read access.

If the file format you specified is a supported file type, then the data in the file may be corrupted.
- Restore the file from a backup copy of the file, if it is available.

Set Not Supported at Run Time

You can only set this property at design time.

Set Not Supported

You tried to assign a value to a property that is read only at design time and run time.

Get Not Supported at Run Time

You can only get a value for this property at design time.

Get Not Supported

You tried to get a value for a property that is write only.

Invalid Clipboard Format

The specified Clipboard format is incompatible with the method that is being executed.
- The two supported Clipboard formats include:

CF_DIB - image data
Wang Annotation - annotation data

Invalid Picture

Only an icon, bitmap, or Windows metafile can be assigned to the **Picture** property or **Picture** object.

- You attempted to assign a file to a **Picture** property or a **Picture** object that is not recognized as an icon, bitmap, or Microsoft Windows metafile.   For the **MouseIcon** property, the valid pictures are .ICO and .CUR file types.

Printer Error

There is a problem that prevents printing.
- You do not have a printer installed.

Open My Computer and double-click on the printer icon to install your printer.   The Printers window opens, and then you can double-click on the Add Printer icon to install your printer.
- The printer is not online.

Switch the printer online.
- The printer is jammed, or out of paper.

Correct the paper jam, or add paper.
- You tried to print a form to a printer that can accept only text.

Switch to an installed printer that can print graphics.

Invalid Property Array Index

The index number is out of bounds for the specified property.
- Ensure that your index number is within the range for the property you are using.

Invalid Icon

You have set the **MouseIcon** property to an invalid file type.
- Valid Icon file types are .ICO and .CUR.

Internal Error

An internal error has occurred.

Cancel Pressed

The **CancelError** property in the Admin OCX has been set to True.   During run time, the Cancel Pressed error will be returned when the user chooses Cancel.   If you do not want this error message to be returned, set the **CancelError** property to False during design time.   If you are using the Scan OCX, the Cancel Pressed error is returned from a method to inform you that a cancel occurred.

Scanner Already Open

The scanner is already open.

- If the scanner is in use by another application, close the scanner in the other application, and retry your operation.   Otherwise, close the scanner in the current application, via the CloseScanner method, and retry your operation.

Scanner Error

An error has occurred with either the scanner or the scanner handler.   This error may be related to your scanner hardware or software.
- Check your scanner hardware to ensure that it is properly powered on, and then retry the scan operation.
- Save your work-in-progress, reboot your PC, and then retry the scan operation.

Bad Size

You selected an invalid image size.
- Select a different image size from the TWAIN setup.

Start Scanner Failure

The start scanner operation failed.
- Close the scanner driver software, and then retry the scanner operation.

Scanner Timeout

The scanner timed out.

- Close the scanner, and then retry the operation.

Scanner Not Open

The scanner is closed.   The scanner must be opened prior to initiating a scan operation.
- Open the scanner using the OpenScanner method, and then retry your operation.

Invalid Registration Information

The Scan registration information is incorrect.
- Edit the registration information or reinstall the scanner and its software.

No Paper Feeder is Available

No paper feeder is available to the scanner.
- Install a paper feeder on your scanner if there is not one already installed.
- If a paper feeder is installed, then check the cable connections between the scanner and the paper feeder.

No Paper

There is no paper in the scanner's paper feeder.
- Check the paper supply and add paper to the feeder.

File Limit

Unable to generate new filenames in the current directory.

- Check the path to ensure that it is valid and retry the template scan.
- Check if all the file numbers have been used.   Valid file numbers range from img00000 through img99999.   If all the file numbers have been used, create a new directory and continue scanning.

Scanner Powered Off

The scanner is powered off.

- Check that the scanner is plugged in and that the power is turned on.

Scanner Cover Open

The scanner cover is open.
- Close the scanner cover and retry your operation.

Scan Aborted

The scan operation was aborted while the scanner was still operating.
- This error message is informational.   No action is necessary.

Scanner Jammed

The scanner is jammed.
- Clear the paper path and retry the scan operation.

Invalid Compression Type

The compression type is invalid.
- Select a valid compression type from the Page Options dialog box.
- Specify a valid compression type for the specified image file type.

Setting	Description
0	Unknown
1	No compression
2	Group3(1D)
3	Group3(Modified Huffman)
4	PackBits
5	Group4(2D)
6	JPEG

AWD and BMP files must be saved using a Compression Type setting of 1.

Invalid Page Type

The page type is invalid.
- Select a different page type from the Page Options dialog box or Scanner Setup dialog box.
- Specify a valid page type based on the specified image file type.

File Types	Valid Page Type
TIF	All except BGR24
AWD	Black and White
BMP	All except RGB24

Invalid Compression Information

The compression information is invalid.

- Select valid compression information from the Page Options dialog box.   You can access the Page Options dialog box from the **Image** property page of the Scan OCX.

The method or property used requires that an image is displayed in the Image Edit control window.
- Verify that an image is displayed in the window prior to calling this method or setting this property.

No Image Specified

The **Image** property for the specified image window has not been set correctly or it is blank.

- Check the **Image** property to ensure that a valid path and image filename have been set.

Invalid Annotation Selected

The specified method cannot be completed using the selected annotation.

- Verify that the selected annotation supports the attribute(s) that the method is trying to alter. For example, if the selected annotation mark is an embedded image, then the **SetSelectedAnnotationLineWidth** method cannot be used.

Parameters Exceed the Maximum Limit

The argument(s) for the specified method exceed the maximum number of characters.

- Respecify the argument(s) using shorter string lengths; for example, the maxiumum length for the **DestImageControl** property is 50 characters.   A valid string length for this property must be 50 characters or less.

Set Not Supported at Design Time

You cannot set this property at design time.   You can only set this property at run time.

No Selection Rectangle Drawn

The specified method or property requires that a selection rectangle must be drawn on the displayed image.

- Before you call the method or before you set the specified property, ensure that a selection rectangle is drawn.
- Ensure that the **SelectRectangle** property of the Image Edit control is set to True to enable the drawing of selection rectangles.

Unable to Create Tool Palette

Unable to create the Image Annotation Tool Palette window.
- Save your work, terminate the application, and then retry this operation.

Additional Parameters Required

The method requires that you specify additional parameter values.

- Ensure that you specified valid values for all of the parameters required by the method you are using.   For example, the **ClipboardCopy** method allows you to optionally specify the coordinates of the rectangle that you want to copy.   If you choose to provide any of the recrtangle coordinate values for this method, then you must provide all of the coordinate values.

Could Not Get Font Attributes

Could not retrieve the font attributes for the selected annotation.
- Verify that the selected annotation supports the font attribute properties. Save your work, exit from the application, and then retry the operation.

Invalid Annotation Type

The **AnnotationType** property contains an invalid value.

- Set the **AnnotationType** property to a valid value.

Tool Palette Already Displayed

You can only have one annotation tool palette open per application.
- Ensure that you have only one annotation tool palette open or close the existing annotation tool palette.

Tool Palette Not Displayed

To use the **Hide** method to hide the annotation tool palette, the tool palette must first be displayed.
- Use the **ShowAnnotationToolPalette** method to display the standard annotation tool palette prior to executing the **Hide** method.

Scanner is Busy

The scanner is in use by another application.

- Close the application that has the scanner in use, and then retry your operation.

Permission Denied

You do not have access rights to read or write to this file.
- From Windows Explorer, right-mouse click on the file, and select Properties from the pop-up menu.   From the Properties dialog box, select the appropriate access rights.

Illegal Function Call

The method has terminated due to invalid parameters or property values.

- Check the input parameters of the function you have called.

Invalid Property Value

A property has been set to an invalid value.

- Set the property to a valid value and then retry your operation.

Page in Use

This error is returned by the **DeletePages** method from the Image Admin OCX.   You cannot delete a page because it is being displayed by either the Thumbnail OCX or the Image Edit control.

- Close all images that are being displayed by either the Thumbnail OCX or the Image Edit control and then retry your operation.

Invalid Display Scale

The display scale may be valid, but not for the currently displayed image

- Respecify the value of the **Zoom** property in the Image Edit control.   You may have to adjust the scale value several times before you find a valid value.

Invalid Rectangle

Either no rectangle was drawn, or the specified rectangle coordinates were invalid or out of range for the displayed image.

- Ensure that you are in the image selection mode and then drag the image within range or use the **DrawSelectionRectangle** method to draw the rectangle.   All X coordinates must be between 0 and the width of the image.   All Y coordinates must be between 0 and the height of the image.

Invalid Display Options

You have requested an operation on an image that cannot be currently performed.

- Review the parameters that you last used to display the image.   Operations might include editing operations such as cut, copy, or paste, or modifications such as a rotation.   Ensure that the parameters are valid values.
- You may have specified valid parameters for the methods being used individually; however, if you combined any editing operations, you may have created invalid display parameters.

Invalid Page

Either the page number is beyond the range of pages in the file or the page number is 0.

- Ensure that the **Page Number** property is set to a value that is valid for the currently displayed image.

No Annotation Selected

The method requires that an annotation is selected before it is executed.

- Before you execute the method, select the annotation mark.   You can select the annotation mark via the Selection Annotation Tool palette with your mouse.   You can also use the **SelectAnnotationGroup** method to select all of the marks for the selected annotation group, or you can bundle a group of annotations by drawing a selection rectangle around them.

Cannot Delete File

An error occured during the delete operation.

- Ensure that the file exists and that you have provided the correct filename.   If you are on a network, the file may be in use by another user or you may not have the appropriate access rights to perform the delete operation.

Wang Image Edit Control

<u>Properties</u>   <u>Methods</u>   <u>Events</u>

The Wang **Image Edit** control adds imaging and image annotation functions to applications that support 32-bit OLE controls.   It enables end users to display, annotate, manipulate, and manage image files.

Image Display

The Image Edit control allows end users to display image files of various types.   Specifically, the control supports the following file types:

- AWD
- BMP
- DCX
- JPG
- PCX
- TIFF

Image Annotation

The control includes several properties, methods, and events that enable programmers to add annotation functions to application programs.   The annotation functions permit end users to annotate displayed images.   The control also includes an <u>Annotation Tool Palette</u> that can be optionally invoked to provide end users with a complete set of annotation tools.

The control supports the annotation types described in the following list:

Annotation Type	Description
Attach-a-Note	Enters text into a background rectangle on an image
Filled Rectangle	Covers a portion of an image
Freehand Line	Draws a freehand line on a section of text or a portion of an image for emphasis
Hollow Rectangle	Places a border around areas of an image for emphasis
Image Embedded	Embeds an actual copy of another image in an image file
Image Reference	Includes another image in an image file by reference (that is, it links to an external file that contains the image)
Select Annotations	Selects annotation marks for deleting, modifying, moving, or resizing
Straight Line	Underlines text, demarcates a section of a page, or draws callout lines
Text	Enters text directly on an image
Text From File	Enters text from a file on an image
Text Stamp	Places a text stamp directly on an image

Image Manipulation

The Image Edit control has properties, methods, and events that enable programmers to add image manipulation functions to application programs.   These functions permit end users to perform the following tasks on displayed images:

- Copy, cut, and paste to the Clipboard
- Flip
- Rotate
- Scale
- Scroll

- Zoom

File Management

The control also includes methods that enable programmers to add image and annotation file management functions to application programs.   These functions permit end users to retrieve, save, print, and delete image and annotation data.

Linking With the Image Edit Control

The Image Edit control is the main Imaging control.   The Wang **Image Annotation Tool Button** and **Image Scan** controls link to Image Edit control to perform several functions.

Image Annotation Tool Button Control

The Image Annotation Tool Button control links to the Image Edit control to permit the annotation of displayed images.   The Image Annotation Tool Button control sends messages to the Image Edit control that set annotation attributes, thereby changing all of the Image Edit annotation attributes to the values set within the Image Annotation Tool Button control.   When the **Draw** method is invoked, the Image Edit control draws the annotation.

Image Scan Control

The Image Scan control links to the Image Edit control to display images while they are being scanned.

Establishing the Link

To link the Image Annotation Tool Button control to the Image Edit control:

1.   Set the **DestImageControl** property of the Image Annotation Tool Button control to the name of the desired Image Edit control.

2.   Set the **ImageControl** property of the Image Edit control to the same Image Edit control name.

To link the Image Scan control to the Image Edit control:

1.   Set the **DestImageControl** property of the Image Scan control to the name of the desired Image Edit control.

2.   Set the **ImageControl** property of the Image Edit control to the same Image Edit control name.

See Also

Image Annotation Tool Button control

Image Scan control

Imaging Concepts

Properties (Image Edit control)

AnnotationBackColor	CompressionInfo	ImageScaleWidth
AnnotationFillColor	CompressionType	ImageWidth
AnnotationFillStyle	DisplayScaleAlgorithm	MouseIcon
AnnotationFont	Enabled	MousePointer
AnnotationFontColor	FileType	Page
AnnotationGroupCount	hWnd	PageCount
AnnotationImage	Image	PageType
AnnotationLineColor	ImageControl	ScrollBars
AnnotationLineStyle	ImageDisplayed	ScrollPositionX
AnnotationLineWidth	ImageHeight	ScrollPositionY
AnnotationStampText	ImageModified	ScrollShortcutsEnabled
AnnotationTextFile	ImagePalette	SelectionRectangle
AnnotationType	ImageResolutionX	StatusCode
AutoRefresh	ImageResolutionY	Zoom
BorderStyle	ImageScaleHeight	

AnnotationGroupCount Property

Returns the number of annotation groups that are on an image page.

Applies To

Image Edit control

Syntax

*object.***AnnotationGroupCount**

Data Type

Integer

Remarks

The AnnotationGroupCount is used as the upper bound value when indexing through all of the annotation groups.

When a new annotation group is created, the AnnotationGroupCount is not incremented until at least one annotation mark is added to the group.

Available at run-time as read-only.

The **Display** method must be invoked prior to reading this property.

See Also

Annotations
Display method
GetAnnotationGroup method
GetAnnotationMarkCount method
SetCurrentAnnotationGroup method

AutoRefresh Property

Returns or sets whether a displayed image is refreshed immediately after changes are made to properties that can affect its visual appearance.

Applies To

Image Edit control

Syntax

*object.***AutoRefresh**[={*True|False*}]

Data Type

Boolean

Setting	Description
True	Refreshes the image immediately when changes are made to display-related properties
False (default)	The image is not refreshed immediately

Remarks

Set this property to True when you want the image to be refreshed every time a property is changed that affects its visual display.

Set this property to False when you do not want the image to be refreshed every time a property is changed that affects its visual display.   The image can be refreshed at any time by invoking the Refresh method, which updates the image using the current settings.

When the AutoRefresh property is set to True, changing the following properties causes the displayed image to be refreshed automatically:

DisplayScaleAlgorithm property

ImagePalette property

ImageResolutionX property

ImageResolutionY property

ScrollBars property

ScrollPositionX property

ScrollPositionY property

Zoom property

See Also

Refresh method

BorderStyle Property

Returns or sets whether the control is displayed with a border.

Applies To

Image Edit control

Syntax

*object.***BorderStyle**[=*value*]

Data Type

Integer (enumerated)

Setting	Description
0	None
1 (default)	Fixed Single

Remarks

Single-line, fixed-size is the only available border style.

Available at design-time only.

CompressionInfo Property

Returns compression option information for the image specified in the **Image** property, and the page specified in the **Page** property.

Applies To

Image Edit control

Syntax

*object.***CompressionInfo**

Data Type

Long

Property settings are expressed using the following bit-wise values:

Setting	Description	Notes
0	No compression options set.	Applicable only to uncompressed image files
1	EOL -- Include/expect End Of Line. Each line is terminated with a standard end-of-line bit.	Not used for JPEG compression
2	Packed Lines -- Byte-align new lines.	Not used for JPEG compression
4	Prefixed EOL -- Include/expect, prefixed End Of Line. Each strip of data is prefixed by a standard end-of-line bit sequence.	Not used for JPEG compression
8	Compressed LTR -- Compressed bit order, Left-To-Right. The bit order for the compressed data is the most significant bit to the least significant bit.	Not used for JPEG compression
16	Expand LTR -- Expanded bit order, Left-To-Right. The bit order for the expanded data is the most significant bit to the least significant bit.	Not used for JPEG compression
32	Negate -- Invert black and white on expansion. Indicates the setting of the Photometric Interpretation field of a TIFF file.	Not used for JPEG compression
64	High Compression/High Quality	Used for JPEG compression only
128	High Compression/Medium Quality	Used for JPEG compression only
256	High Compression/Low Quality	Used for JPEG compression only
512	Medium Compression/High Quality	Used for JPEG compression only
1024	Medium Compression/Medium Quality	Used for JPEG compression only
2048	Medium Compression/Low Quality	Used for JPEG

		compression only
4096	Low Compression/High Quality	Used for JPEG compression only
8192	Low Compression/Medium Quality	Used for JPEG compression only
16384	Low Compression/Low Quality	Used for JPEG compression only

Remarks

Image files with compression types other than JPEG can have CompressionInfo values between 1 and 63, in any combination.   Image files with the JPEG compression type can have only one specific value between 64 and 16384 (as indicated in the preceding list).

The CompressionInfo value is independent of an image being displayed.   If the image specified in the Image and Page properties is being displayed, the value indicates the compression of the displayed image.   If the image specified in the Image and Page properties is not being displayed, the value indicates the compression of the image specified in the Image property.

Available at run-time as read-only.

See Also

Compression

CompressionType property

Image property

Page property

SaveAs method

CompressionType Property

Returns the compression type of the image specified in the **Image** property, and the page specified in the **Page** property.

Applies To

Image Edit control

Syntax

*object.*__CompressionType__

Data Type

Integer (enumerated)

Property settings are expressed using the following values:

Setting	Description
0	Unknown
1	No compression
2	Group3(1D)
3	Group3(Modified Huffman)
4	PackBits
5	Group4(2D)
6	JPEG

Remarks

The CompressionType value is independent of an image being displayed.   If the image specified in the Image and Page properties is being displayed, the value indicates the compression type of the displayed image.   If the image specified in the Image and Page properties is not being displayed, the value indicates the compression type of the image specified in the Image property.

Available at run-time as read-only.

See Also

Compression

CompressionInfo property

Image property

Page property

SaveAs method

DisplayScaleAlgorithm Property

Returns or sets the scaling algorithm used when displaying images.

Applies To

Image Edit control

Syntax

*object.***DisplayScaleAlgorithm**[*=integer*]

Data Type

Integer (enumerated)

Property settings are expressed using the following values:

Setting	Description	Notes
0 (default)	Normal	Normal decimation
1	Gray4	4-bit gray-scale, 16 shades of gray
2	Gray8	8-bit gray-scale, 256 shades of gray
3	Stamp	Used for thumbnail representation of images
4	Optimize	Changes the display algorithm based on the image type of the displayed image

Remarks

The DisplayScaleAlgorithm value can be specified before or after an image is displayed.   If the value is specified after an image is displayed, the displayed image may change visually.

When the **AutoRefresh** property is set to True, changing the DisplayScaleAlgorithm property causes the displayed image to be refreshed automatically.   Some settings may not be apparent until the image is scaled to a Zoom value under 100 percent.

See Also

AutoRefresh property

Display

Image property

Zoom property

FileType Property

Returns the file type of the image file specified in the **Image** property, and the page specified in the **Page** property.

Applies To

Image Edit control

Syntax

*object.***FileType**

Data Type

Integer (enumerated)

Property settings are expressed using the following values:

Setting	Description
1	TIFF
2	AWD
3	BMP
4	PCX
5	DCX
6	JPG

Remarks

The FileType value is independent of an image being displayed.   If the image specified in the Image and Page properties is being displayed, the value indicates the file type of the displayed image.   If the image specified in the Image and Page properties is not being displayed, the value indicates the file type of the image specified in the Image property.

Available at run-time as read-only.

See Also

Image property

Images

Page property

SaveAs method

Image Property

Returns the fully-qualified filename of an image file being displayed, or sets the fully-qualified filename of an image file to be displayed or otherwise operated on by the Image Edit control.

Applies To

Image Edit control

Syntax

*object.***Image**[*=filename*]

Data Type

String

Remarks

The name must be a fully-qualified filename.   DOS and Windows filenames are permissible.

See Also

Display method

Images

ImageControl Property

Returns or sets the unique identifier to be specified for linking with another Imaging control.   The Wang **Image Annotation Tool Button** and **Image Scan** controls link to the Image Edit control in order to perform several functions.

Applies To

Image Edit control

Syntax

object.**ImageControl**[=*controlname*]

Data Type

String

Remarks

A unique default name is provided when you draw the control on a form.

See Also

DestImageControl property (Image Annotation Tool Button control)

DestImageControl property (Image Scan control)

Image Annotation Tool Button control

Image Scan control

Images

ImageDisplayed Property

Indicates whether an image is currently being displayed.

Applies To

Image Edit control

Syntax

object.**ImageDisplayed**

Data Type

Boolean

Setting	Description
True	An image is being displayed
False	An image is not being displayed

Remarks

Available at run-time as read-only.

See Also

Display method

Images

ImageHeight Property

Returns the height in full-size coordinates of the image being displayed.

Applies To

Image Edit control

Syntax

*object.***ImageHeight**

Data Type

Long

Remarks

If an image is not being displayed, the ImageHeight property returns the full-size height of the image specified in the **Image** property.

If an image is not being displayed and the Image property does not contain the filename of an image, the ImageHeight property generates an error.

Available at run-time as read-only.

See Also

Image property

Images

ImageScaleHeight property

ImageWidth property

ImageModified Property

Indicates whether an image has been modified.

Applies To

Image Edit control

Syntax

*object.***ImageModified**

Data Type

Boolean

Setting	Description
True	The image has been modified
False	The image has not been modified

Remarks

Because the value of this property indicates whether an image file has been modified, you can use it to determine whether your program should prompt the end user to save the image file.

Available at run-time as read-only.

Changes to the following properties set the ImageModified property to True:

ImageResolutionX property

ImageResolutionY property

Invoking the following methods sets the ImageModifed property to True:

AddAnnotationGroup method

BurnInAnnotations method

ClipboardCut method

ClipboardPaste method

ConvertPageType method

DeleteAnnotationGroup method

DeleteImageData method

DeleteSelectedAnnotations method

Draw method

EditSelectedAnnotationText method

Flip method

HideAnnotationGroup method

RotateLeft method

RotateRight method

SetSelectedAnnotationBackColor method

SetSelectedAnnotationFillColor method

SetSelectedAnnotationFillStyle method

SetSelectedAnnotationFont method

SetSelectedAnnotationFontColor method

SetSelectedAnnotationLineColor method

SetSelectedAnnotationLineStyle method

SetSelectedAnnotationLineWidth method

ShowAnnotationGroup method

Drawing an annotation using the standard Annotation Tool Palette sets the ImageModifed property to True.

See Also

Close event

Images

Save method

SaveAs method

ImagePalette Property

Returns or sets the palette to use when displaying an image.

Applies To

Image Edit control

Syntax

*object.***ImagePalette**[=*value*]

Data Type

Integer (enumerated)

Property settings are expressed using the following values:

Setting	Description	Notes
0 (default)	Custom	Selects a custom palette
1	Common	Selects the common palette
2	Gray8	Selects the 8-bit gray-scale, 256 shades-of-gray palette
3	RGB24	Selects the 24-bit, millions-of-colors palette
4	Black-and-white	Selects the black-and-white palette

Remarks

The ImagePalette value can be specified before or after an image is displayed.   If the value is specified after an image is displayed, the displayed image may change visually.

When the **AutoRefresh** property is set to True, changing the ImagePalette property causes the displayed image to be refreshed automatically.

See Also

AutoRefresh property

Images

Palettes

SetImagePalette method

ImageResolutionX Property

Returns or sets the horizontal resolution of the image being displayed.

Applies To

Image Edit control

Syntax

object.**ImageResolutionX**[*=integer*]

Data Type

Long

Remarks

When the **AutoRefresh** property is set to True, changing the ImageResolutionX property causes the displayed image to be refreshed automatically.

If an image is not being displayed, the ImageResolutionX property returns or sets the horizontal resolution of the image specified in the **Image** property.

If an image is not being displayed and the Image property does not contain the filename of an image, the ImageResolutionX property generates an error.

Available at run-time as read-and-write.

See Also

AutoRefresh property

Image property

ImageResolutionY property

Images

ImageResolutionY Property

Returns or sets the vertical resolution of the image being displayed.

Applies To

Image Edit control

Syntax

object.**ImageResolutionY**[*=integer*]

Data Type

Long

Remarks

When the **AutoRefresh** property is set to True, changing the ImageResolutionY property causes the displayed image to be refreshed automatically.

If an image is not being displayed, the ImageResolutionY property returns or sets the vertical resolution of the image specified in the **Image** property.

If an image is not being displayed and the Image property does not contain the filename of an image, the ImageResolutionY property generates an error.

Available at run-time as read-and-write.

See Also

AutoRefresh property

Image property

ImageResolutionX property

Images

ImageScaleHeight Property

Returns the height in pixels of the image as it is displayed.

Applies To

Image Edit control

Syntax

*object.***ImageScaleHeight**

Data Type

OLE_YSIZE_PIXELS

Remarks

The **Display** method must be invoked prior to using this property.

The ImageScaleHeight property differs from the **ImageHeight** property.   The ImageHeight property returns the height in full-size coordinates, while the ImageScaleHeight property value is affected by the current zoom factor.

The ImageScaleHeight and ImageHeight properties also differ for images with asymmetric resolutions.   For example, when an 8½- by 11-inch image with a resolution of 200 by 100 dots per inch (dpi) is displayed, the ImageHeight property returns a value of 1100, while the ImageScaleHeight property returns a value of 2200 due to pixel replication that occurs at display-time to optimize display quality.

Available at run-time as read-only.

See Also

Display method

Image property

ImageHeight property

ImageResolutionX property

ImageResolutionY property

Images

ImageScaleWidth property

Zoom property

ImageScaleWidth Property

Returns the width in pixels of the image as it is displayed.

Applies To

Image Edit control

Syntax

*object.***ImageScaleWidth**

Data Type

OLE_XSIZE_PIXELS

Remarks

The **Display** method must be invoked prior to using this property.

The ImageScaleWidth property differs from the **ImageWidth** property.   The ImageWidth property returns the width in full-size coordinates, while the ImageScaleWidth property value is affected by the current zoom factor.

The ImageScaleWidth and ImageWidth properties also differ for images with asymmetric resolutions.   For example, when an 8½- by 11-inch image with a resolution of 100 by 200 dots per inch (dpi) is displayed, the ImageWidth property returns a value of 850, while the ImageScaleWidth property returns a value of 1700 due to pixel replication that occurs at display-time to optimize display quality.

Available at run-time as read-only.

See Also

Display method

Image property

ImageResolutionX property

ImageResolutionY property

Images

ImageScaleHeight property

ImageWidth property

Zoom property

ImageWidth Property

Returns the width in full-size coordinates of the image being displayed.

Applies To

Image Edit control

Syntax

*object.***ImageWidth**

Data Type

Long

Remarks

If an image is not being displayed, the ImageWidth property returns the full-size width of the image specified in the **Image** property.

If an image is not being displayed and the Image property does not contain the filename of an image, the ImageWidth property generates an error.

Available at run-time as read-only.

See Also

Image property

ImageHeight property

Images

ImageScaleWidth property

MouseIcon Property

Returns or sets the mouse icon to be displayed when the mouse pointer is moved over an Image Edit control.

Applies To

Image Edit control

Syntax

*object.***MouseIcon**=**LoadPicture**(*pathname*)

Data Type

String

The pathname is a string expression specifying the path and filename of the file containing the custom icon (.CUR or .ICO files).

Remarks

The MouseIcon property provides a custom icon that is used when the **MousePointer** property is set to 99.

See Also

MousePointer property

MousePointer Property

Returns or sets the type of mouse pointer to be displayed when the pointer is moved over an Image Edit control at run-time.

Applies To

Image Edit control

Syntax

*object.***MousePointer**[*=integer*]

Data Type

Integer (enumerated)

Setting	Description
0 (Default)	As determined by the value of the **AnnotationType** property
1	Arrow
2	Cross (cross-hair pointer)
3	I-Beam
4	Arrow
5	Size (four-pointed arrow pointing North, South, East, and West)
6	Size NE SW (double arrow pointing NorthEast and SouthWest)
7	Size N S (double arrow pointing North and South)
8	Size NW SE (double arrow pointing NorthWest and SouthEast)
9	Size W E (double arrow pointing West and East)
10	Up Arrow
11	Hourglass (wait)
12	No Drop
13	Arrow and Hourglass
14	Arrow and Question mark
15	Size All
16	Freehand Line
17	Hollow Rectangle
18	Filled Rectangle
19	Text Stamp
20	Text
21	Text From File
22	Attach-a-Note
23	Hand
24	Image Selection
99	Custom pointer specified by the **MouseIcon** property.

Remarks

The MouseIcon property must be set prior to setting the MousePointer property to 99.

Use this property to indicate changes in functionality as the mouse pointer passes over controls on a form or dialog box.   Use the Hourglass setting (11) to indicate that the end user should wait for a process or operation to finish.

See Also

AnnotationType property

MouseIcon property

Page Property

Returns the page number of an image file where an action was performed, or sets the page number of an image file where an action is to be performed.

Applies To

Image Edit control

Syntax

*object.***Page**[*=integer*]

Data Type

Long

Remarks

The default value for this property is 1.   This property is typically used to specify the page to be displayed by the **Display** method.

See Also

Display method

Image Page

PageCount property

PageCount Property

Returns the total number of pages in the image file that is currently being displayed.

Applies To

Image Edit control

Syntax

*object.***PageCount**

Data Type

Long

Remarks

If an image is not being displayed, the PageCount property returns the total number of pages in the image file specified in the **Image** property.

If an image is not being displayed and the Image property does not contain the filename of an image, the PageCount property generates an error.

Available at run-time as read-only.

See Also

Image Page

Image property

Page property

PageType Property

Returns the image type of the image specified in the **Image** property and the page specified in the **Page** property.

Applies To

Image Edit control

Syntax

*object.***PageType**

Data Type

Integer (enumerated)

Property settings are expressed using the following values:

Setting	Description
1	Black-and-white
2	Gray4
3	Gray8
4	Palettized4
5	Palettized8
6	RGB24
7	BGR24

Remarks

The PageType property value is independent of an image being displayed.   If the image page specified in the Page property is being displayed, the value is the image type of the displayed image.   If the image page specified in the Image and Page properties is not being displayed, the value is the image type of the page specified in the Image and Page properties.

Available at run-time as read-only.

See Also

ConvertPageType method

Image Page

Page property

SaveAs method

ScrollBars Property

Returns or sets whether a displayed image contains horizontal and/or vertical scroll bars.

Applies To

Image Edit control

Syntax

*object.***ScrollBars**[={*True|False*}]

Data Type

Boolean

Setting	Description
True (default)	Horizontal and/or vertical scroll bars are displayed
False	No scroll bars are displayed

Remarks

If this property is set to True, horizontal and/or vertical scroll bars are displayed only when the image extends beyond the object's borders.

If the image has scroll bars and the **ScrollShortcutsEnabled** property is set to True, accelerator keys are provided to permit the end user to scroll an image using the keyboard.

When the **AutoRefresh** property is set to True, changing the ScrollBars property causes the displayed image to be refreshed automatically.

See Also

AutoRefresh property

Images

Scroll event

ScrollImage method

ScrollShortcutsEnabled property

ScrollPositionX property

ScrollPositionY property

ScrollPositionX Property

Returns or sets the horizontal scroll position in pixels.

Applies To

Image Edit control

Syntax

*object.***ScrollPositionX**[*=integer*]

Data Type

Long

Remarks

Setting the ScrollPositionX property causes the displayed image to scroll horizontally to the pixel position specified.

To set the scroll position prior to displaying an image, the ScrollPositionX and **ScrollPositionY** properties must be set in the **Load** event.

When the image is first displayed using the **Display** method, this property is set to 0 by default.

When the **AutoRefresh** property is set to True, changing the ScrollPositionX property causes the displayed image to be refreshed automatically.

Available at run-time as read-and-write.

See Also

AutoRefresh property
Display method
Images
Load event
Scroll event
ScrollBars property
ScrollImage method
ScrollPositionY property

ScrollPositionY Property

Returns or sets the vertical scroll position in pixels.

Applies To

Image Edit control

Syntax

*object.***ScrollPositionY**[*=integer*]

Data Type

Long

Remarks

Setting the ScrollPositionY property causes the displayed image to scroll vertically to the pixel position specified.

To set the scroll position prior to displaying an image, the **ScrollPositionX** and ScrollPositionY properties must be set in the **Load** event.

When the image is first displayed using the **Display** method, this property is set to 0 by default.

When the **AutoRefresh** property is set to True, changing the ScrollPositionY property causes the displayed image to be refreshed automatically.

Available at run-time as read-and-write.

See Also

AutoRefresh property

Display method

Images

Load event

Scroll event

ScrollBars property

ScrollImage method

ScrollPositionX property

ScrollShortcutsEnabled Property

Returns or sets whether the object recognizes a set of accelerator keys for scrolling a displayed image provided horizontal and/or vertical scroll bars are present.

Applies To

Image Edit control

Syntax

*object.***ScrollShortcutsEnabled**[={*True*|*False*}]

Data Type

Boolean

Setting	Description
True (default)	Accelerator keys can be used to scroll an image
False	Accelerator keys cannot be used

Remarks

If this property and the **ScrollBars** property are set to True, and the image extends beyond the object's borders, the end user can use accelerator keys to scroll the displayed image.

Under certain circumstances, it may be necessary to set the focus to the Image Edit control for the accelerator keys to function.

The following accelerator keys provide scrolling capabilities:

Accelerator Key	Description
Page Down	Scrolls the image downward by one full screen
Page Up	Scrolls the image upward by one full screen
Ctrl + Page Down	Scrolls the image to the right by one full screen
Ctrl + Page Up	Scrolls the image to the left by one full screen
Right Arrow	Scrolls the image to the right by 4 pixels
Left Arrow	Scrolls the image to the left by 4 pixels
Up Arrow	Scrolls the image upward by 4 pixels
Down Arrow	Scrolls the image downward by 4 pixels

See Also

Images

Scroll event

ScrollBars property

ScrollImage method

SelectionRectangle Property

Indicates whether a selection box will be drawn when the end user clicks the left mouse button and drags the mouse pointer over a displayed image.

Applies To

Image Edit control

Syntax

*object.***SelectionRectangle**[=*boolean*]

Data Type

Boolean

Setting	Description
True (default)	Selection box drawing is enabled
False	Selection box drawing is not enabled

Remarks

Once the end user draws a selection box, cutting and copying to the Clipboard can be accomplished.

See Also

ClipboardCopy method

ClipboardCut method

SelectionRectDrawn event

ZoomToSelection method

StatusCode Property

Returns a status code, which identifies the error that occurred when the last property was set or read, or the last method was invoked.   A status code of 0 indicates that no error has occurred.

Applies To

Image Admin control, Image Annotation Tool Button control, Image Edit control, Image Scan control, Image Thumbnail control

Syntax

*object.***StatusCode**

Data Type

Long

Remarks

The default value for this property is 0.

Available at run-time as read-only.

Zoom Property

Returns or sets the zoom factor for an image.

Applies To

Image Edit control

Syntax

object.**Zoom**[*=value*]

Data Type

Single

Remarks

The Zoom value is expressed as a percentage.   The valid range is from 2 to 6500 percent; the default value is 100 percent.

When invoked, the **FitTo** and **ZoomToSelection** methods update the value of the Zoom property.

A Zoom value can be specified before or after an image is displayed.   If the image currently specified in the **Image** property is being displayed when the Zoom value is changed, the displayed image may change visually.

When the **AutoRefresh** property is set to True, changing the Zoom property causes the displayed image to be refreshed automatically.

See Also

AutoRefresh property

Display

FitTo method

Image property

ZoomToSelection method

Methods (Image Edit control)

AboutBox	GetAnnotationMarkCount	ScrollImage
AddAnnotationGroup	GetCurrentAnnotationGroup	SelectAnnotationGroup
BurnInAnnotations	GetSelectedAnnotationBackColor	SelectTool
ClearDisplay	GetSelectedAnnotationFillColor	SetCurrentAnnotationGroup
ClipboardCopy	GetSelectedAnnotationFillStyle	SetImagePalette
ClipboardCut	GetSelectedAnnotationFont	SetSelectedAnnotationBackColor
ClipboardPaste	GetSelectedAnnotationFontColor	SetSelectedAnnotationFillColor
CompletePaste	GetSelectedAnnotationImage	SetSelectedAnnotationFillStyle
ConvertPageType	GetSelectedAnnotationLineColor	SetSelectedAnnotationFont
DeleteAnnotationGroup	GetSelectedAnnotationLineStyle	SetSelectedAnnotationFontColor
DeleteImageData	GetSelectedAnnotationLineWidth	SetSelectedAnnotationLineColor
DeleteSelectedAnnotations	HideAnnotationGroup	SetSelectedAnnotationLineStyle
Display	HideAnnotationToolPalette	SetSelectedAnnotationLineWidth
DisplayBlankImage	IsClipboardDataAvailable	ShowAnnotationGroup
Draw	PrintImage	ShowAnnotationToolPalette
DrawSelectionRect	Refresh	ShowAttribsDialog
EditSelectedAnnotationText	RotateLeft	ShowRubberStampDialog
FitTo	RotateRight	ZoomToSelection
Flip	Save	
GetAnnotationGroup	SaveAs	

AboutBox Method

Displays an About dialog box that contains the version number of the control.

Applies To

Image Admin control, Image Annotation Tool Button control, Image Edit control, Image Scan control, Image Thumbnail control

Syntax

object.**AboutBox**

Arguments

None

Return Value

None

AddAnnotationGroup Method

Adds a new annotation group to an image page.

Applies To

Image Edit control

Syntax

*object.***AddAnnotationGroup** *GroupName*

Argument

Parameter	Data Type	Setting
GroupName	String	The name of the new annotation group

Return Value

None

Remarks

The **Display** method must be invoked prior to calling this method.

The annotation group is not actually created until the first annotation mark is placed on the image.

See Also

Annotations

DeleteAnnotationGroup method

Display method

Images

BurnInAnnotations Method

Burns annotations onto an image, causing them to be permanently incorporated into the image.   Use this method with caution because once annotations are burned-in, they can no longer be removed or modified.

Applies To

Image Edit control

Syntax

*object.***BurnInAnnotations** *Option,MarkOption*[*,GroupName*]

Arguments

Parameter	Data Type	Setting	Description
Option	Integer (enumerated)	0	All annotations marks, invisible and visible, are saved to the base image
		1	All annotations that are visible are saved to the base image
		2	All annotations that are selected are saved to the base image
MarkOption (Refer to the Notes section following this list for more information.)	Integer (enumerated)	0	Changes annotations to black (on black-and-white images only)
		1	Changes annotations to white (on black-and-white images only)
		2	Default rendering -- Does not change the colors of annotations for non-bilevel images (for example, color and gray-scale images)
GroupName	String	Name	The name of the group that contains the annotations (if specified, the Option parameter is ignored)

Notes Regarding the Use of the MarkOption Parameter

MarkOption Set to 0

For black-and-white image types, the BurnInAnnotations method with a MarkOption of 0 changes annotations to black.

MarkOption Set to 1

For black-and-white image types, the BurnInAnnotations method with a MarkOption of 1 changes annotations to white.

MarkOption Set to 2

For black-and-white image types, the BurnInAnnotations method with a MarkOption of 2 changes annotations to black, white, or black-and-white dither patterns based on the type of annotation mark, its color, and its transparency setting.   Refer to the following list:

Annotation Type	MarkOption2 Burn-In Format
Attach-a-Note	Burned-in as a black or white rectangle based on the original background color, with text burned-in as black or white based on the original color of the font. Light colors, like yellow, are burned-in as white; dark colors, like red, are burned-in as black. The underlying image data is obscured.
Freehand Line, Hollow Rectangle, Straight Line, Text, Text From File, and Text Stamp	Burned-in as black or white based on the original color of the annotation. Light colors, like yellow, are burned-in as white; dark colors, like red, are burned-in as black.
Filled Rectangle	Burned-in as black-and-white dithered rectangles. The underlying image data is obscured.
Highlighter	Burned-in as black-and-white transparent dithered rectangles. The underlying image data is visible.
Image Embedded and Image Reference	Burned-in as a black-and-white representation of the original image.

For all other image types, the BurnInAnnotations method with a MarkOption of 2 burns annotations onto the image using the displayed colors.   Then it converts the image to a 24-bit color image (BGR24 or RGB24).   Because 24-bit color images require a large amount of memory and disk space, use caution when selecting this option.   To minimize the memory and disk space requirements of 24-bit color images, you can use the **SaveAs** method to save the 24-bit color images as 8-bit Palettized images.

To save color annotations on a black-and-white image, use the **ConvertPageType** method to convert the image to a page type other than black-and-white.   Then invoke the BurnInAnnotations method with a Mark Option of 2; the BurnInAnnotations method burns the annotations onto the image in color.

Return Value

None

Remarks

After invoking the BurnInAnnotations method, the **Save** or SaveAs method must be invoked to save the combined image and annotation.

When a GroupName is specified, the Option parameter is ignored and all annotation marks in the specified GroupName are burned-in.

See Also

Annotations

ConvertPageType method

Display method

Images

Save method

SaveAs method

ClearDisplay Method

Clears a displayed image in the control.

Applies To

Image Edit control

Syntax

object.**ClearDisplay**

Arguments

None

Return Value

None

Remarks

An image must be displayed before the ClearDisplay method can be invoked.

The ClearDisplay method does not save any changes that may have been made to an existing image, nor does it change the value of the **Image** property.   It does, however, trigger the **Close** event.

See Also

Close event

Display

Image property

ClipboardCopy Method

Copies the selected or specified portion of an image or annotation to the Clipboard.

Applies To

Image Edit control

Syntax

object.**ClipboardCopy** [*Left,Top,Width,Height*]

Arguments

Parameter	Data Type	Setting
Left	Long	The upper-left corner of the selection rectangle in pixel coordinates of the displayed image
Top	Long	The top of the selection rectangle in pixel coordinates of the displayed image
Width	Long	The width of the selection rectangle in pixels
Height	Long	The height of the selection rectangle in pixels

Return Value

None

Remarks

If the ClipboardCopy method is invoked without parameters after an end user draws a selection rectangle, the image data bounded by the selection rectangle is copied to the Clipboard.

If the ClipboardCopy method is invoked without parameters and without an end user drawing a selection rectangle, the ClipboardCopy method generates an error.

To copy annotations to the Clipboard, the end user selects the desired annotation marks using the Select Annotations tool.

To copy image data to the Clipboard, the end user selects the desired area using the image selection rectangle, which is the default when no annotation tool is selected.

It is not possible to copy both image and annotation data to the Clipboard simultaneously.

See Also

ClipboardCut method

ClipboardPaste method

Display

DrawSelectionRect method

SelectionRectangle property

SelectionRectDrawn event

ClipboardCut Method

Copies the selected or specified portion of an image or annotation to the Clipboard and then clears (cuts) the portion from the image or annotation.

Applies To

Image Edit control

Syntax

*object.***ClipboardCut** [*Left,Top,Width,Height*]

Arguments

Parameter	Data Type	Setting
Left	Long	The upper-left corner of the selection rectangle in pixel coordinates of the displayed image
Top	Long	The top of the selection rectangle in pixel coordinates of the displayed image
Width	Long	The width of the selection rectangle in pixels
Height	Long	The height of the selection rectangle in pixels

Return Value

None

Remarks

If the ClipboardCut method is invoked without parameters after an end user draws a selection rectangle, the image data bounded by the selection rectangle is copied to the Clipboard and cleared from the image.

If the ClipboardCut method is invoked without parameters and without an end user drawing a selection rectangle, the ClipboardCut method generates an error.

To cut annotations to the Clipboard, the end user selects the desired annotation marks using the Select Annotations tool.

To cut image data to the Clipboard, the end user selects the desired area using the image selection rectangle, which is the default when no annotation tool is selected.

It is not possible to cut both image and annotation data to the Clipboard simultaneously.

See Also

ClipboardCopy method

ClipboardPaste method

Display

DrawSelectionRect method

SelectionRectangle property

SelectionRectDrawn event

ClipboardPaste Method

Pastes image or annotation data from the Clipboard.

Applies To

Image Edit control

Syntax

*object.***ClipboardPaste** [*Left,Top*]

Arguments

Parameter	Data Type	Setting
Left	Long	The left point of the displayed image in pixels
Top	Long	The top point of the displayed image in pixels

Return Value

None

Remarks

To determine if data is available for pasting from the Clipboard, invoke the **IsClipboardDataAvailable** method.

If the Left and Top parameters are not entered, the data is pasted at the left-top point of the displayed image.

See Also

ClipboardCopy method

ClipboardCut method

CompletePaste method

Display

IsClipboardDataAvailable method

PasteCompleted event

CompletePaste Method

Completes a Clipboard paste operation, making the pasted image or annotation data part of the original image.

Applies To

Image Edit control

Syntax

object.**CompletePaste**

Arguments

None

Return Value

None

Remarks

The **ClipboardPaste** method pastes image or annotation data from the Clipboard into the control. After the image or annotation data appears in the control, it can be moved at will and is therefore not part of the original image.   To make the pasted image or annotation data part of the original image, the Clipboard paste operation must be completed.   When invoked, the CompletePaste method completes a Clipboard paste operation.

A Clipboard paste operation can also be completed by an end user clicking on the form, outside of the pasted data.

See Also

ClipboardPaste method

PasteCompleted event

ConvertPageType Method

Converts a displayed image to the page type specified and optionally repaints the image.

Applies To

Image Edit control

Syntax

object.**ConvertPageType** *PageType*[,*Repaint*]

Arguments

Parameter	Data Type	Setting
PageType	Integer (enumerated)	The desired <u>PageType</u> setting
Repaint	Boolean	True -- Image is repainted (default) False -- Image is not repainted

Return Value

None

Remarks

An image must be displayed before the ConvertPageType method can be invoked.   If the Repaint parameter is not specified, True is entered as a default (the image is repainted).   Invoking this method sets the **ImageModified** property to True.

To save the converted image, the **Save** or **SaveAs** method must be invoked.

An image cannot be converted to a PageType of Palettized4 unless it presently has a 4-bit palette.

See Also

<u>ImageModified property</u>

<u>Images</u>

<u>PageType property</u>

<u>Save method</u>

<u>SaveAs method</u>

DeleteAnnotationGroup Method

Deletes an annotation group as well as its annotation marks, and then redisplays the image.

Applies To

Image Edit control

Syntax

*object.***DeleteAnnotationGroup** *GroupName*

Argument

Parameter	Data Type	Setting
GroupName	String	The name of the annotation group to delete

Return Value

None

Remarks

The **Display** method must be invoked prior to calling this method.

To save the image without the annotation marks, the **Save** or **SaveAs** method must be invoked.

See Also

AddAnnotationGroup method

Annotations

DeleteSelectedAnnotations method

Display method

Save method

SaveAs method

DeleteImageData Method

Deletes a selected or specified portion of an image.

Applies To

Image Edit control

Syntax

*object.***DeleteImageData** [*Left,Top,Width,Height*]

Arguments

Parameter	Data Type	Setting
Left	Long	The upper-left corner of the selection rectangle in pixel coordinates of the displayed image
Top	Long	The top of the selection rectangle in pixel coordinates of the displayed image
Width	Long	The width of the selection rectangle in pixels
Height	Long	The height of the selection rectangle in pixels

Return Value

None

Remarks

If the DeleteImageData method is invoked without parameters after an end user draws a selection rectangle, the area bounded by the selection rectangle is deleted.   This method does not delete annotation data.

If the DeleteImageData method is invoked without parameters and without an end user drawing a selection rectangle, the DeleteImageData method generates an error.

After invoking the DeleteImageData method, the **Save** or **SaveAs** method must be invoked to save the altered image.

See Also

DeleteSelectedAnnotations method

DrawSelectionRect method

Images

Save method

SaveAs method

SelectionRectangle property

SelectionRectDrawn event

DeleteSelectedAnnotations Method

Deletes selected annotations.

Applies To

Image Edit control

Syntax

object.**DeleteSelectedAnnotations**

Arguments

None

Return Value

None

Remarks

To be deleted, annotations must first be selected using the Select Annotations tool.   If no annotations are selected, the DeleteSelectedAnnotations method generates an error.

After invoking the DeleteSelectedAnnotations method, the **Save** or **SaveAs** method must be invoked to save the altered image.

Note that annotations can also be deleted by invoking the **DeleteAnnotationGroup** method.

See Also

Annotations

DeleteAnnotationGroup method

Save method

SaveAs method

Display Method

Displays the image specified in the **Image** property.

Applies To

Image Edit control

Syntax

*object.***Display**

Arguments

None

Return Value

None

Remarks

The Display method triggers a **Load** event before the actual image is displayed.   End users can then hide or show any of the annotation groups before they are actually displayed, or set any other display-oriented properties (for example, Zoom, ScrollPosition, etc.).   If an image is already displayed, this method triggers a **Close** event.

See Also

Close event

Display

Load event

Image property

ImageDisplayed property

Page property

DisplayBlankImage Method

Displays a blank image according to the dimensions specified.

Applies To

Image Edit control

Syntax

object.**DisplayBlankImage** *ImageWidth,ImageHeight*[*,ResolutionX,ResolutionY,PageType*]

Arguments

Parameter	Data Type	Setting
ImageWidth	Long	The width of the image in pixels at full-scale (100 percent)
ImageHeight	Long	The height of the image in pixels at full-scale (100 percent)
ResolutionX	Long	The horizontal resolution of the image in dots per inch
ResolutionY	Long	The vertical resolution of the image in dots per inch
PageType	Integer (enumerated)	The desired <u>PageType</u> setting

Return Value

None

Remarks

When the DisplayBlankImage method is invoked with an image already displayed, it triggers a **Close** event.   The displayed image file is closed without being saved, and the new image data is displayed.   The value of the **Image** property remains unchanged.

If the ResolutionX parameter or the ResolutionY parameter is not entered, a value of 200 is used as a default.   If the PageType parameter is not entered, the Black-and-White page type is used as a default.

See Also

<u>Close event</u>

<u>Display</u>

DrawSelectionRect Method

Draws a selection rectangle on an image.

Applies To

Image Edit control

Syntax

*object.***DrawSelectionRect** *Left,Top,Width,Height*

Arguments

Parameter	Data Type	Setting
Left	Long	The beginning horizontal position of the rectangle relative to the image window in pixels
Top	Long	The beginning vertical position of the rectangle relative to the image window in pixels
Width	Long	The width of the selection rectangle in pixels
Height	Long	The height of the selection rectangle in pixels

Return Value

None

Remarks

None

See Also

ClipboardCopy method
ClipboardCut method
DeleteImageData method
Images
SelectionRectangle property
SelectionRectDrawn event
ZoomToSelection method

EditSelectedAnnotationText Method

Displays a dialog box with the selected text annotation.   The dialog box enables the end user to modify the text.

Applies To

Image Edit control

Syntax

*object.***EditSelectedAnnotationText** *Left,Top*

Arguments

Parameter	Data Type	Setting
Left	Long	The beginning horizontal position of the mark to be edited
Top	Long	The beginning vertical position of the mark to be edited

Return Value

None

Remarks

The left and top coordinates of the annotation to be edited can be obtained from the **MarkSelect** event.

After invoking the EditSelectedAnnotationText method, the **Save** or **SaveAs** method must be invoked to save the altered annotation.

See Also

Annotations

MarkSelect event

Save method

SaveAs method

FitTo Method

Scales the image relative to the image window.

Applies To

Image Edit control

Syntax

*object.***FitTo** *Option*[*,Repaint*]

Arguments

Parameter	Data Type	Setting	Description
Option	Integer (enumerated)	0 1 2 3	Best fit -- Fits the image to the size of the window Fit to width -- Fits the image to the width of the window Fit to height -- Fits the image to the height of the window Inch to inch -- Maintains the relative size of the original image regardless of the resolution of the image
Repaint	Boolean	True False	Image is repainted (default) Image is not repainted

Return Value

None

Remarks

The Inch to inch setting calculates the size of the image dependent upon the dots per inch reported by the display driver.

When invoked, the FitTo method updates the **Zoom** property with the new zoom factor.

The FitTo method can be invoked before or after an image is displayed.   If an image is displayed when the method is invoked, and the Repaint parameter is set to True, the image is redisplayed using the Option setting specified.

See Also

Display

Zoom property

ZoomToSelection method

Flip Method

Rotates a displayed image by 180 degrees.

Applies To

Image Edit control

Syntax

object.**Flip**

Arguments

None

Return Value

None

Remarks

The **Display** method must be invoked prior to calling this method.

Invoking the Flip method modifies the displayed image.   The **Save** or **SaveAs** method must be invoked to save the modified image.

See Also

Display

Display method

RotateLeft method

RotateRight method

Save method

SaveAs method

GetAnnotationGroup Method

Returns the name of the annotation group based on the index specified.

Applies To

Image Edit control

Syntax

*object.***GetAnnotationGroup**(*Index*)

Argument

Parameter	Data Type	Setting
Index	Integer	The index value of the annotation group

Return Value

String

Remarks

The **Display** method must be invoked prior to calling this method.

Index is a 0-based value, where 0 is the first annotation group in the array, 1 is the second, 2 is the third, and so on.   The upper bound value of the index can be obtained from the **AnnotationGroupCount** property.

The GetAnnotationGroup method is valid only at run-time.

See Also

AnnotationGroupCount property

Annotations

Display method

GetCurrentAnnotationGroup method

HideAnnotationGroup method

SelectAnnotationGroup method

SetCurrentAnnotationGroup method

ShowAnnotationGroup method

GetAnnotationMarkCount Method

Returns the number of annotation marks on an image page or in an annotation group.

Applies To

Image Edit control

Syntax

*object.***GetAnnotationMarkCount**([*GroupName,AnnotationType*])

Arguments

Parameter	Data Type	Setting
GroupName	String	The name of the annotation group
AnnotationType	Integer (enumerated)	The desired <u>AnnotationType</u> setting

Return Value

Integer

Remarks

The **Display** method must be invoked prior to calling this method.

If the GroupName and Annotation Type parameters are both entered, the GetAnnotationMarkCount method returns the number of annotation marks of the type specified that are in the annotation group specified.

If the GroupName parameter is entered, but the Annotation Type is not entered, the GetAnnotationMarkCount method returns the total number of annotation marks that are in the annotation group specified.

If the GroupName parameter is not entered, but the Annotation Type is entered, the GetAnnotationMarkCount method returns the number of annotation marks of the type specified that are in all of the annotation groups.

If no parameters are entered, the GetAnnotationMarkCount method returns the total number of annotation marks that are on the page.

The GetAnnotationMarkCount method is valid only at run-time.

See Also

<u>AnnotationGroupCount property</u>

<u>Annotations</u>

<u>AnnotationType property</u>

<u>Display method</u>

GetCurrentAnnotationGroup Method

Returns the name of the annotation group to which subsequent annotations will belong.

Applies To

Image Edit control

Syntax

*object.***GetCurrentAnnotationGroup**

Arguments

None

Return Value

String

Remarks

The GetCurrentAnnotationGroup method is valid only at run-time.

See Also

Annotations

BurnInAnnotations method

GetAnnotationGroup method

HideAnnotationGroup method

SetCurrentAnnotationGroup method

SelectAnnotationGroup method

ShowAnnotationGroup method

GetSelectedAnnotationBackColor Method

Returns the background color of a selected Attach-a-Note annotation.

Applies To

Image Edit control

Syntax

*object.***GetSelectedAnnotationBackColor**

Arguments

None

Return Value

Color

Remarks

The GetSelectedAnnotationBackColor method is valid only at run-time.   It is used with annotations of the Attach-a-Note type.

Color is expressed using the Red-Green-Blue (R,G,B) format.

See Also

AnnotationBackColor property

Annotations

AnnotationType property

SetSelectedAnnotationBackColor method

GetSelectedAnnotationFillColor Method

Returns the color used to fill a selected Filled Rectangle annotation.

Applies To

Image Edit control

Syntax

*object.***GetSelectedAnnotationFillColor**

Arguments

None

Return Value

Color

Remarks

The GetSelectedAnnotationFillColor method is valid only at run-time. It is used with annotations of the Filled Rectangle type.

Color is expressed using the Red-Green-Blue (R,G,B) format.

See Also

AnnotationFillColor property

Annotations

AnnotationType property

SetSelectedAnnotationFillColor method

GetSelectedAnnotationFillStyle Method

Returns the style used to fill a selected image or Filled Rectangle annotation.

Applies To

Image Edit control

Syntax

*object.***GetSelectedAnnotationFillStyle**

Arguments

None

Return Value

Integer (enumerated)

Value	Description
0	Transparent
1	Opaque

Remarks

The GetSelectedAnnotationFillStyle method is valid only at run-time.   It is used with annotations of the following types:

Filled Rectangle

Image Embedded

Image Reference

See Also

AnnotationFillStyle property

Annotations

AnnotationType property

SetSelectedAnnotationFillStyle method

GetSelectedAnnotationFont Method

Returns a font object's properties.   Applies to selected text annotations.

Applies To

Image Edit control

Syntax

*object.***GetSelectedAnnotationFont**

Arguments

None

Return Value

Font object

Remarks

For example, the following code returns the properties of the font object identified by the GetSelectedAnnotationFont method of an Image Edit object:

```
Dim x As Font
Set x = Imgedit1.GetSelectedAnnotationFont
CommonDialog1.FontBold = x.Bold
CommonDialog1.FontItalic = x.Italic
CommonDialog1.FontName = x.Name
CommonDialog1.FontSize = x.SIZE
CommonDialog1.FontUnderline = x.Underline
CommonDialog1.Font Strikethru = x.StrikeThrough
CommonDialog1.Flags = &H103&
CommonDialog1.ShowFont
```

The GetSelectedAnnotationFont method is valid only at run-time.   It is used with annotations of the following types:

Text

Attach-a-Note

Text From File

Text Stamp

See Also

AnnotationFont property

Annotations

AnnotationType property

SetSelectedAnnotationFont method

GetSelectedAnnotationFontColor Method

Returns the color used in a selected text annotation.

Applies To

Image Edit control

Syntax

*object.***GetSelectedAnnotationFontColor**

Arguments

None

Return Value

Color

Remarks

The GetSelectedAnnotationFontColor method is valid only at run-time.   It is used with annotations of the following types:

Text

Attach-a-Note

Text From File

Text Stamp

Color is expressed using the Red-Green-Blue (R,G,B) format.

See Also

AnnotationFontColor property

Annotations

AnnotationType property

SetSelectedAnnotationFontColor method

GetSelectedAnnotationImage Method

Returns the fully-qualified filename of the image that is being used in a selected image annotation.

Applies To

Image Edit control

Syntax

*object.***GetSelectedAnnotationImage**

Arguments

None

Return Value

String

Remarks

The GetSelectedAnnotationImage method is valid only at run-time.   It is used with annotations of the following types:

Image Embedded

Image Reference

See Also

AnnotationImage property

Annotations

AnnotationType property

Images

GetSelectedAnnotationLineColor Method

Returns the line color used in a selected line or Hollow Rectangle annotation.

Applies To

Image Edit control

Syntax

*object.***GetSelectedAnnotationLineColor**

Arguments

None

Return Value

Color

Remarks

The GetSelectedAnnotationLineColor method is valid only at run-time.   It is used with annotations of the following types:

 Freehand Line

 Hollow Rectangle

 Straight Line

Color is expressed using the Red-Green-Blue (R,G,B) format.

See Also

AnnotationLineColor property

Annotations

AnnotationType property

SetSelectedAnnotationLineColor method

GetSelectedAnnotationLineStyle Method

Returns the line style used in a selected line or Hollow Rectangle annotation.

Applies To

Image Edit control

Syntax

*object.***GetSelectedAnnotationLineStyle**

Arguments

None

Return Value

Integer (enumerated)

Value	Description
0	Transparent
1	Opaque

Remarks

The GetSelectedAnnotationLineStyle method is valid only at run-time.   It is used with annotations of the following types:

Freehand Line

Hollow Rectangle

Straight Line

See Also

AnnotationLineStyle property

Annotations

AnnotationType property

SetSelectedAnnotationLineStyle method

GetSelectedAnnotationLineWidth Method

Returns the line width in pixels used in a selected line or Hollow Rectangle annotation.

Applies To

Image Edit control

Syntax

*object.***GetSelectedAnnotationLineWidth**

Arguments

None

Return Value

Integer

Remarks

The GetSelectedAnnotationLineWidth method is valid only at run-time.   It is used with annotations of the following types:

Freehand Line

Hollow Rectangle

Straight Line

See Also

AnnotationLineWidth property

Annotations

AnnotationType property

SetSelectedAnnotationLineWidth method

HideAnnotationGroup Method

Hides an annotation group.

Applies To

Image Edit control

Syntax

*object.***HideAnnotationGroup** [*GroupName*]

Argument

Parameter	Data Type	Setting
GroupName	String	The name of the annotation group to hide

Return Value

None

Remarks

The **Display** method must be invoked prior to calling this method.

If the GroupName parameter is not specified, the HideAnnotationGroup method hides all annotation groups.

See Also

Annotations

Display method

GetAnnotationGroup method

GetCurrentAnnotationGroup method

SetCurrentAnnotationGroup method

ShowAnnotationGroup method

HideAnnotationToolPalette Method

Hides the Annotation Tool Palette.

Applies To

Image Edit control

Syntax

*object.***HideAnnotationToolPalette**

Arguments

None

Return Value

None

Remarks

The **ShowAnnotationToolPalette** method must be invoked prior to calling this method.

See Also

Annotations

ShowAnnotationToolPalette method

ToolPaletteHidden event

IsClipboardDataAvailable Method

Determines whether image or annotation data is present in the Clipboard.

Applies To

Image Edit control

Syntax

*object.***IsClipboardDataAvailable**

Arguments

None

Return Value

Boolean

Value	Description
True	Image or annotation data is in the clipboard
False	Image or annotation data is not in the clipboard

Remarks

Use the IsClipboardDataAvailable method to determine whether the **ClipboardPaste** method can be invoked.

See Also

Annotations

ClipboardCopy method

ClipboardCut method

ClipboardPaste method

Images

PrintImage Method

Prints the image being displayed.

Applies To

Image Edit control

Syntax

object.**PrintImage** [*StartPage,EndPage,OutputFormat,Annotations,Printer,Driver,PortNumber*]

Arguments

Parameter	Data Type	Setting
StartPage	Long	The starting page in a range of pages to be printed
EndPage	Long	The ending page in a range of pages to be printed
OutputFormat	Integer (enumerated)	The output format to be used: 0 -- Pixel to pixel 1 -- Inch to inch 2 -- Fit to page
Annotations	Boolean	Whether annotations will be printed with the image: True -- Annotations will be printed False -- Annotations will not be printed
Printer	String	The name of the printer
Driver	String	The name of the printer driver
PortNumber	String	The name of the output port

Return Value

None

Remarks

If an image is displayed when the PrintImage method is invoked, the image is printed as it appears, including any changes made.

If an image is not being displayed, the PrintImage method prints the image specified in the **Image** property.   If an image is not being displayed and the Image property does not contain the filename of an image, the PrintImage method generates an error.

If the StartPage and EndPage parameters are not entered, this method prints all pages by default.

If the Annotations parameter is not specified, all annotations are printed by default.

See Also

Image property

Images

Refresh Method

Repaints the contents of the Image Edit control.

Applies To

Image Edit control

Syntax

*object.***Refresh**

Arguments

None

Return Value

None

Remarks

The **Display** method must be invoked prior to calling this method.

See Also

Display method

DisplayScaleAlgorithm property

ImagePalette property

ImageResolutionX property

ImageResolutionY property

ScrollBars property

ScrollPositionX property

ScrollPositionY property

Zoom property

RotateLeft Method

Rotates the displayed image 90 degrees to the left.

Applies To

Image Edit control

Syntax

*object.***RotateLeft**

Arguments

None

Return Value

None

Remarks

The **Display** method must be invoked prior to calling this method.

Invoking the RotateLeft method modifies the displayed image.   The **Save** or **SaveAs** method must be invoked to save the modified image.

See Also

Display
Display method
Flip method
RotateRight method
Save method
SaveAs method

RotateRight Method

Rotates the displayed image 90 degrees to the right.

Applies To

Image Edit control

Syntax

*object.***RotateRight**

Arguments

None

Return Value

None

Remarks

The **Display** method must be invoked prior to calling this method.

Invoking the RotateRight method modifies the displayed image.   The **Save** or **SaveAs** method must be invoked to save the modified image.

See Also

Display
Display method
Flip method
RotateLeft method
Save method
SaveAs method

Save Method

Saves the displayed image (page) to the original path and file.

Applies To

Image Edit control

Syntax

*object.***Save** [*SaveAtZoom*]

Argument

Parameter	Data Type	Setting
SaveAtZoom	Boolean	Whether the image is saved using the current or original scale percentage value:
		True -- The image is saved using the current scale percentage value, as specified in the Zoom property
		False -- The image is saved using the original scale percentage value (Zoom value of 100 percent)

Return Value

None

Remarks

The **Display** method must be invoked prior to calling this method.

For the TIFF file type, annotations are saved as annotation data.   For all other file types, annotations are lost unless the **BurnInAnnotations** method was invoked prior to calling the Save method.   The BurnInAnnotations method burns annotations onto an image, causing them to be permanently incorporated into the image.

The Save method saves the image to the same file from which it was opened.

If the SaveAtZoom parameter is not specified, the image is saved using the original zoom scale percentage value.

See Also

Annotations
BurnInAnnotations method
Close event
Display
Display method
ImageModified property
Images
SaveAs method
Zoom property

SaveAs Method

Saves the displayed image using the filename specified.

Applies To

Image Edit control

Syntax

object.**SaveAs** *Image*[,*FileType,PageType,CompressionType,CompressionInfo,SaveAtZoom*]

Arguments

Parameter	Data Type	Setting
Image	String	The name of the image file being saved
FileType	Integer (enumerated)	The FileType the image is being saved as
PageType	Integer (enumerated)	The PageType the image is being saved as
CompressionType	Integer (enumerated)	The CompressionType at which the image is being saved
CompressionInfo	Integer (enumerated)	The CompressionInfo setting of the image being saved
SaveAtZoom	Boolean	Whether the image is being saved using the current or original scale percentage value:
		True -- The image is being saved using the current scale percentage value False -- The image is being saved using the original scale percentage value

Return Value

None

Remarks

The **Display** method must be invoked prior to calling this method.

The SaveAs method saves the image using the filename specified in the Image parameter.   If the file already exists, it is overwritten.

Upon successful completion of the SaveAs method, the image display window continues to display the original image; the **Image** property value remains unchanged.   Continuing to display the original image ensures that the image on display is the same as the image specified in the Image property.   For example, assume the currently displayed image is **payroll.awd** and the value of the Image property is also **payroll.awd**.   When the SaveAs method is invoked with its Image parameter set to **payroll96.awd**, the image is saved as **payroll96.awd**.   The image display window continues to display **payroll.awd**, and the Image property value remains unchanged at **payroll.awd**.

If a multipage image file (AWD, DCX, or TIFF) is saved as a file type that does not support multipage image files (for example, BMP), only the page currently being displayed is saved to the new file.

For the TIFF file type, annotations are saved as annotation data.   For all other file types, annotations are lost unless the **BurnInAnnotations** method is invoked prior to calling the SaveAs method.   The BurnInAnnotations method burns annotations onto an image, causing them to be permanently incorporated into the image.

If the FileType parameter is not entered, the SaveAs method uses the original FileType setting of the image as a default.   If the original FileType setting cannot be determined, the method saves the image as a TIFF file.

If the PageType and FileType parameters are not entered, the SaveAs method uses the original PageType setting of the image as a default.   However, if a FileType is entered, the SaveAs method uses a PageType default setting that is associated with the FileType setting of the image.   The following list describes the default PageType settings:

FileType Setting	Default PageType Setting
1 -- TIFF	PageType setting of the displayed image
2 -- AWD	1 -- Black-and-White
3 -- BMP	PageType setting of the displayed image

If the CompressionType parameter is not entered, the SaveAs method uses a default CompressionType setting that is associated with the FileType and PageType settings of the image. The following list describes the default CompressionType settings:

FileType Setting	PageType Setting	Default CompressionType Setting
2 -- AWD	1 -- Black-and-white	1 -- No compression
3 -- BMP	1,5,7 -- All supported PageTypes	1 -- No compression
1 -- TIFF	1 -- Black-and-white	2 -- Group3(1D)
1 -- TIFF	2 -- Gray4	1 -- No compression
1 -- TIFF	3 -- Gray8	6 -- JPEG
1 -- TIFF	4 -- Palettized4	1 -- No compression
1 -- TIFF	5 -- Palettized8	1 -- No compression
1 -- TIFF	6 -- RGB24	6 -- JPEG

Note: To use a PageType setting of 4 (Palettized4), the displayed image must already be a Palettized4 image.

If the CompressionInfo parameter is not entered, the SaveAs method uses a default CompressionInfo setting that is associated with the CompressionType setting of the image.   The following list describes the default CompressionInfo settings:

CompressionType Setting	CompressionInfo Default Setting
1 -- No compression	0 -- No compression
2 -- Group3(1D)	8 -- Compressed LTR, 16 -- Expand LTR, 1 -- EOL, and 4 -- Prefixed EOL
3 --Group3(Modified Huffman)	8 -- Compressed LTR and 16 -- Expand LTR
4 -- PackBits	16 -- Expand LTR
5 -- Group4(2D)	2 -- Packed Lines, 8 -- Compressed LTR, and

16 -- Expand LTR

6 -- JPEG   1024 -- Medium Compression/Medium Quality

When a CompressionInfo value of 0 is entered with the CompressionType parameter set to 1 (no compression), the file is saved using no compression.

When a CompressionInfo value of 0 is entered with the CompressionType parameter set to a value other than 1, the file is saved using the default CompressionInfo parameter settings that correspond to the setting of the CompressionType parameter (as explained in the preceding list).

For example, if the end user specifies a CompressionType value of 1 (no compression) and then a CompressionInfo value of 0, the file is saved using no compression.   If the end user specifies a CompressionType value of 2 (Group3[1D]) and then a CompressionInfo value of 0, the file is saved using the default CompressionInfo values for Group3(1D):

8 -- Compressed LTR,
16 -- Expand LTR,
1 -- EOL, and
4 -- Prefixed EOL

If the SaveAtZoom parameter is not specified, the image is saved using the original zoom scale percentage value.

See Also

BurnInAnnotations method

Compression

CompressionInfo property

CompressionType property

Close event

Display

Display method

FileType property

Image property

ImageModified property

Images

PageType property

Save method

ScrollImage Method

Scrolls a displayed image.

Applies To

Image Edit control

Syntax

*object.***ScrollImage** *Direction,ScrollAmount*

Arguments

Parameter	Data Type	Setting
Direction	Integer (enumerated)	The direction in which to scroll the image: 0 -- Down 1 -- Up 2 -- Right 3 -- Left
ScrollAmount	Long	The number of pixels to scroll the image

Return Value

None

Remarks

The **Display** method must be invoked prior to calling this method.

See Also

Display method

Images

Scroll event

ScrollBars property

ScrollPositionX property

ScrollPositionY property

ScrollShortcutsEnabled property

SelectAnnotationGroup Method

Selects all annotation marks within a specific group on the image page.

Applies To

Image Edit control

Syntax

*object.***SelectAnnotationGroup** *GroupName*

Argument

Parameter	Data Type	Setting
GroupName	String	The name of the annotation group to select

Return Value

None

Remarks

The **Display** method must be invoked prior to calling this method.

See Also

AddAnnotationGroup method

Annotations

Display method

GetAnnotationGroup method

GetCurrentAnnotationGroup method

MarkSelect event

SetCurrentAnnotationGroup method

ShowAnnotationGroup method

SelectTool Method

Selects an annotation tool from the Annotation Tool Palette.

Applies To

Image Edit control

Syntax

*object.***SelectTool** *ToolID*

Argument

Parameter	Data Type	Setting	Description
ToolID	Integer (enumerated)	0	No Tool
		1	Selection
		2	Freehand Line
		3	Highlighter
		4	Straight Line
		5	Hollow Rectangle
		6	Filled Rectangle
		7	Text
		8	Attach-a-Note
		9	Text From File
		10	Text Stamp

Return Value

None

Remarks

The Highlighter annotation type is composed of a Filled Rectangle annotation type with a transparent fill style.

When a tool is selected from the tool palette, the previous tool is deselected automatically.

The Annotation Tool Palette does not have to be visible to use this method.

See Also

Annotations

ShowAnnotationToolPalette method

ToolSelected event

SetCurrentAnnotationGroup Method

Sets the annotation group to which subsequent annotations will belong.

Applies To

Image Edit control

Syntax

*object.***SetCurrentAnnotationGroup** *GroupName*

Argument

Parameter	Data Type	Setting
GroupName	String	The name of the annotation group

Return Value

None

Remarks

The SetCurrentAnnotationGroup method is valid only at run-time.

The names of the groups can be obtained from the **GetAnnotationGroup** method.

A new group can be created using the **AddAnnotationGroup** method.

See Also

AddAnnotationGroup method

Annotations

BurnInAnnotations method

GetAnnotationGroup method

GetCurrentAnnotationGroup method

SetImagePalette Method

Sets the palette to be used for the image currently being displayed, or an image to be displayed.

Applies To

Image Edit control

Syntax

*object.***SetImagePalette** *Option*

Argument

Parameter	Data Type	Setting	Description
Option	Integer (enumerated)	0 1	Foreground Background

Return Value

None

Remarks

Images are displayed using the foreground palette by default.   When the default foreground palette is used, it is loaded into the Windows palette, which can cause other images or other color-aware objects to become distorted.   Setting the SetImagePalette Option value to 1 (Background) sets the existing palette, which has already been loaded by Windows.

See Also

ImagePalette property

Images

Palettes

SetSelectedAnnotationBackColor Method

Sets the background color of a selected Attach-a-Note annotation.

Applies To

Image Edit control

Syntax

*object.***SetSelectedAnnotationBackColor** *Color*

Argument

Parameter	Data Type	Setting
Color	Color	The desired Attach-a-Note color

Return Value

None

Remarks

The SetSelectedAnnotationBackColor method is used with annotations of the Attach-a-Note type.

Use the RGB format.   To ensure that the text in an Attach-a-Note annotation will be visible against the background, specify a background color that is significantly different from the font color set using the **SetSelectedAnnotationFontColor** method.

See Also

AnnotationBackColor property

AnnotationType property

Annotations

GetSelectedAnnotationBackColor method

SetSelectedAnnotationFontColor method

SetSelectedAnnotationFillColor Method

Sets the color used to fill a selected Filled Rectangle annotation.

Applies To

Image Edit control

Syntax

*object.***SetSelectedAnnotationFillColor** *Color*

Argument

Parameter	Data Type	Setting
Color	Color	The desired Filled Rectangle color

Return Value

None

Remarks

The SetSelectedAnnotationFillColor method is used with annotations of the Filled Rectangle annotation type.

Use the RGB format.

See Also

AnnotationFillColor property

Annotations

AnnotationType property

GetSelectedAnnotationFillColor method

SetSelectedAnnotationFontColor method

SetSelectedAnnotationFillStyle Method

Sets the style used to fill a selected image or Filled Rectangle annotation.

Applies To

Image Edit control

Syntax

*object.***SetSelectedAnnotationFillStyle** *Style*

Argument

Parameter	Data Type	Setting	Description
Style	Integer (enumerated)	0 1	Transparent -- The underlying image data is visible Opaque -- The underlying image data is obscured

Return Value

None

Remarks

The SetSelectedAnnotationFillStyle method is used with annotations of the following types:

Filled Rectangle

Image Embedded

Image Reference

See Also

AnnotationFillStyle property

Annotations

AnnotationType property

GetSelectedAnnotationFillStyle method

SetSelectedAnnotationFont Method

Sets a font object's properties.   Applies to selected text annotations.

Applies To

Image Edit control

Syntax

*object.***SetSelectedAnnotationFont** *Font*

Argument

Parameter	Data Type	Setting
Font	Font object	The properties of the font object

Return Value

None

Remarks

Use the SetSelectedAnnotationFont method to identify the specific font object whose properties you want to set.   The following list shows the properties you can set, along with their data types, descriptions, and defaults:

Property	Data Type	Description	Default
Bold	Boolean	True -- Selects bold attribute False -- Deselects bold attribute	False
Italic	Boolean	True -- Selects italic attribute False -- Deselects italic attribute	False
Name	String	The desired font name	MS San Serif
Size	Single	The desired font size in points (0 to 2048)	12
Underline	Boolean	True -- Selects underline attribute False -- Deselects underline attribute	False
StrikeThrough	Boolean	True -- Selects ~~strikethrough~~ attribute False -- Deselects strikethrough attribute	False

For example, the following code sets the properties of the font object identified by the SetSelectedAnnotationFont method of an Image Edit object:

```
Dim x As Font
Set x = Form1.Font
x.Bold = True
x.Italic = True
x.Name = "Courier New"
x.SIZE = 14
x.Underline = True
x.StrikeThrough = True
Imgedit1.SetSelectedAnnotationFont x
```

The SetSelectedAnnotationFont method is used with annotations of the following types:

Text

Attach-a-Note

Text From File

Text Stamp

See Also

SetSelectedAnnotationFontColor Method

Sets the font color used in a selected text annotation.

Applies To

Image Edit control

Syntax

*object.***SetSelectedAnnotationFontColor** *Color*

Argument

Parameter	Data Type	Setting
Color	Color	The desired font color

Return Value

None

Remarks

The SetSelectedAnnotationFontColor method is used with annotations of the following types:

Text

Attach-a-Note

Text From File

Text Stamp

Use the RGB format.   To ensure that the text in an Attach-a-Note annotation will be visible against the background, specify a font color that is significantly different from the background color set using the **SetSelectedAnnotationBackColor** method.

See Also

AnnotationFontColor property

Annotations

AnnotationType property

GetSelectedAnnotationFontColor method

SetSelectedAnnotationBackColor method

SetSelectedAnnotationFillColor method

SetSelectedAnnotationLineColor Method

Sets the line color used in a selected line or Hollow Rectangle annotation.

Applies To

Image Edit control

Syntax

*object.***SetSelectedAnnotationLineColor** *Color*

Argument

Parameter	Data Type	Setting
Color	Color	The desired line color

Return Value

None

Remarks

The SetSelectedAnnotationLineColor method is used with annotations of the following types:

Freehand Line

Hollow Rectangle

Straight Line

Use the RGB format.

See Also

AnnotationLineColor property

Annotations

AnnotationType property

GetSelectedAnnotationLineColor method

SetSelectedAnnotationLineStyle Method

Sets the line style used in a selected line or Hollow Rectangle annotation.

Applies To

Image Edit control

Syntax

*object.***SetSelectedAnnotationLineStyle** *Style*

Argument

Parameter	Data Type	Setting	Description
Style	Integer (enumerated)	0 1	Transparent -- The underlying image data is visible Opaque -- The underlying image data is obscured

Return Value

None

Remarks

The SetSelectedAnnotationLineStyle method is used with annotations of the following types:

Freehand Line

Hollow Rectangle

Straight Line

See Also

AnnotationLineStyle property

Annotations

AnnotationType property

GetSelectedAnnotationLineStyle method

SetSelectedAnnotationLineWidth Method

Sets the line width used in a selected line or Hollow Rectangle annotation.

Applies To

Image Edit control

Syntax

*object.***SetSelectedAnnotationLineWidth** *Width*

Argument

Parameter	Data Type	Setting
Width	Integer	The desired line width in pixels

Return Value

None

Remarks

The SetSelectedAnnotationLineWidth method is used with annotations of the following types:

Freehand Line

Hollow Rectangle

Straight Line

See Also

AnnotationLineWidth property

Annotations

AnnotationType property

GetSelectedAnnotationLineWidth method

ShowAnnotationGroup Method

Shows the specified annotation group.

Applies To

Image Edit control

Syntax

*object.***ShowAnnotationGroup** [*GroupName*]

Argument

Parameter	Data Type	Setting
GroupName	String	The name of the desired annotation group

Return Value

None

Remarks

The **Display** method must be invoked prior to calling this method.

If the GroupName parameter is not entered, the ShowAnnotationGroup method shows all annotation groups.

See Also

Annotations

Display method

GetAnnotationGroup method

GetCurrentAnnotationGroup method

HideAnnotationGroup method

SelectAnnotationGroup method

ShowAnnotationToolPalette Method

Shows the Annotation Tool Palette, which enables end users to edit images displayed in this control.

Applies To

Image Edit control

Syntax

*object.***ShowAnnotationToolPalette** [*ShowAttributesDialog,Left,Top,ToolTipText*]

Arguments

Parameter	Data Type	Setting
ShowAttributesDialog	Boolean	Whether the Annotation Attributes dialog box is invoked:
		True -- When the end user clicks on a tool palette button using the right mouse button and selects properties from a pop-up menu, the Annotation Attributes dialog box appears. When the end user clicks on the Rubber Stamp tool button, a pop-up menu appears that contains a list of previously defined stamps.
		False -- The Annotation Attributes dialog box cannot be invoked.
Left	Long	The horizontal position of the Annotation Tool Palette in pixels, relative to the upper-left corner of the Image Edit control.
Top	Long	The vertical position of the Annotation Tool Palette in pixels, relative to the upper-left corner of the Image Edit control.
ToolTipText	String	Contains the tool tip text used for each Annotation Tool Palette button. Ten strings are required; use the vertical bar (|) character to separate each string. If not specified, default strings are provided.

Return Value

None

Remarks

The ShowAnnotationToolPalette method can be invoked only once per control.

If the ShowAttributesDialog parameter is not entered, it is set to True by default.

If the Left and Top parameters are not entered, the Annotation Tool Palette is displayed at Position 10,10 relative to the Image Edit control.

ToolTip strings are displayed by default.

To draw annotations using the Annotation Tool Palette, an annotation tool must be selected.   This can be accomplished by the end user clicking on a button on the tool palette, or by invoking the **SelectTool** method.   The annotation mark can then be drawn manually by the end user, or programatically by calling the **Draw** method.   Changing the **AnnotationType** property with the Annotation Tool Palette displayed will not achieve the desired results.

See Also

ShowAttribsDialog Method

Shows an Annotation Attributes dialog box, which the end user can use to change the attributes of the selected annotation mark.

Applies To

Image Edit control

Syntax

*object.***ShowAttribsDialog**

Arguments

None

Return Value

None

Remarks

An annotation must be selected before invoking the ShowAttribsDialog method.

See Also

Annotations

SelectAnnotationGroup method

SelectTool method

ShowRubberStampDialog Method

Can be used in conjunction with the Annotation Tool Palette to show the Rubber Stamp Properties dialog box to allow the end user to create, delete, and edit Rubber Stamps.

Applies To

Image Edit control

Syntax

*object.***ShowRubberStampDialog**

Arguments

None

Return Value

None

Remarks

None

See Also

Annotations

SelectTool method

ShowAnnotationToolPalette method

ZoomToSelection Method

Scales a selected portion of a displayed image so that it fits into the size of the control.

Applies To

Image Edit control

Syntax

*object.*ZoomToSelection

Arguments

None

Return Value

None

Remarks

A selection rectangle must be drawn on the image prior to calling the ZoomToSelection method.

The ZoomToSelection method updates the **Zoom** property with the zoom factor used.

See Also

Display

Display method

DrawSelectionRect method

FitTo method

SelectionRectDrawn event

Zoom property

Events (Image Edit control)

The following events used by the Image Edit control are all standard:

Click	KeyUp
DblClick	MouseDown
Error	MouseMove
KeyDown	MouseUp
KeyPress	

The **Error** event occurs only when an error condition triggers it.   It does not occur when a property is set or a method is invoked.

If you are using Visual Basic, it adds the **DragDrop**, **DragOver**, **GotFocus**, and **LostFocus** events.

The following list shows the custom, Image Edit control events:

Close	Scroll
Load	SelectionRectDrawn
MarkEnd	ToolPaletteHidden
MarkSelect	ToolTip
PasteCompleted	ToolSelected

Close Event

This event occurs whenever one of the following methods is invoked:
- **Display**
- **ClearDisplay**
- **DisplayBlankImage**

The Close event can be used to indicate to the end user that the displayed image is about to be closed.

Applies To

Image Edit control

Syntax

Sub *object*_**Close**( )

Arguments

None

Remarks

Once a Close event occurs, any changes made to displayed images are lost unless the **Save** or **SaveAs** method is invoked.

If desired, your program can check the **ImageModified** property to see if any changes have been made to the displayed image.   If changes have been made, the program can be written so it prompts the end user to save the changes.

See Also

ClearDisplay method

Display

Display method

DisplayBlankImage method

ImageModified property

Save method

SaveAs method

Load Event

This event occurs immediately after the **Display** method is invoked.

Applies To

Image Edit control

Syntax

Sub *object*_**Load**(*Zoom*)

Argument

Parameter	Data Type	Setting
Zoom	Double	Returns the zoom factor used when the image was displayed by the Display method.
		The AWD file type is the only one that stores the actual zoom factor used. Therefore, the Zoom parameter returns the actual zoom factor only when images of the AWD file type are displayed. All other file types cause the Zoom parameter to return the default zoom factor of 100.
		The zoom factor is typically expressed as a percentage. The valid range is from 2 to 6500 percent; the default value is 100 percent.
		The Zoom parameter can also return the following values:
		-1 -- Fit to width. The image was fit to the width of the window when it was displayed. -2 -- Fit to height. The image was fit to the height of the window when it was displayed. -3 -- No previously saved zoom factor.
		Note: Zoom values of -1 and -2 may not be compatible with some applications.

Remarks

Once a Load event occurs, the visibility of the group can be changed, and other properties and methods can be set or invoked.

If desired, your program can use the returned zoom factor to set the **Zoom** property, or the Option parameter of the **FitTo** method, prior to displaying another image.

See Also

Display

Display method

FitTo method

Zoom property

MarkEnd Event

This event occurs immediately after the end user or program completes the drawing of an annotation mark.

Applies To

Image Edit control

Syntax

Sub *object*_**MarkEnd**(*Left,Top,Width,Height,MarkType,GroupName*)

Arguments

Parameter	Data Type	Setting
Left	Long	The starting horizontal position of the annotation mark within the image in image pixels
Top	Long	The starting vertical position of the annotation mark within the image in image pixels
Width	Long	The ending horizontal position of the annotation mark within the image in image pixels
Height	Long	The ending vertical position of the annotation mark within the image in image pixels
MarkType	Integer	The AnnotationType of the annotation mark (except for the Select Annotations annotation type)
GroupName	String	The name of the annotation group to which the annotation belongs

Remarks

The MarkType parameter cannot be the Select Annotations annotation type.

A MarkEnd event does not occur if a Select Annotation rectangle is drawn (AnnotationType of 0).

See Also

Annotations

Draw method

PasteCompleted Event

This event occurs immediately after pasted image or annotation data is committed to a location on an image.   Pasted image or annotation data is committed when an end user clicks on a form, outside of a pasted image or annotation, or whenever the **CompletePaste** method is invoked.

Applies To

Image Edit control

Syntax

Sub *object*_**PasteCompleted**( )

Arguments

None

Remarks

The **ClipboardPaste** method must be invoked for this event to be triggered.

See Also

ClipboardPaste method

CompletePaste method

Scroll Event

This event occurs immediately after the end user or program scrolls an image in the control.

Applies To

Image Edit control

Syntax

Sub *object*_**Scroll**( )

Arguments

None

Remarks

None

See Also

ScrollBars property

ScrollImage method

ScrollShortcutsEnabled property

ScrollPositionX property

ScrollPositionY property

MarkSelect Event

This event occurs immediately after the end user or program selects an annotation mark.

Applies To

Image Edit control

Syntax

Sub *object*_**MarkSelect**(*Button,Shift,Left,Top,Width,Height,MarkType,GroupName*)

Arguments

Parameter	Data Type	Setting
Button	Integer (enumerated)	Specifies the mouse button pressed: 1 -- Left button 2 -- Right button 4 -- Middle button
Shift	Integer (enumerated)	Specifies whether the Shift, Ctrl, or Alt key is pressed: 1 -- Shift key 2 -- Ctrl key 4 -- Alt key
Left	Long	The beginning horizontal position within the image in image pixels
Top	Long	The beginning vertical position within the image in image pixels
Width	Long	The ending horizontal position within the image in image pixels
Height	Long	The ending vertical position within the image in image pixels
MarkType	Integer	The AnnotationType of the annotation mark (except for the Select Annotations annotation type)
GroupName	String	The name of the annotation group to which the annotation belongs

Remarks

If multiple annotations are selected, a MarkType parameter value of 0 is returned.

The GroupName parameter is an empty string when multiple annotations are selected.

See Also

Annotations

SelectAnnotationGroup method

SelectionRectDrawn Event

This event occurs immediately after the end user presses the left mouse button, or the **DrawSelectionRect** method is invoked.

Applies To

Image Edit control

Syntax

Sub *object*_**SelectionRectDrawn**(*Left,Top,Width,Height*)

Arguments

Parameter	Data Type	Setting
Left	Long	The beginning horizontal position within the image in image pixels relative to the upper-left corner of the Image Edit display window
Top	Long	The beginning vertical position within the image in image pixels relative to the upper-left corner of the Image Edit display window
Width	Long	The ending horizontal position within the image in image pixels relative to the upper-left corner of the Image Edit display window
Height	Long	The ending vertical position within the image in image pixels relative to the upper-left corner of the Image Edit display window

Remarks

The **SelectionRectangle** property must be set to True for this event to be triggered.

If the end user draws a selection rectangle of 3 pixels or less, no selection rectangle is drawn. When this happens, the SelectionRectDrawn event is still triggered, but all of its parameters contain a value of 0.   Your program can use these 0 values to determine whether cut, copy, or delete operations should be enabled.

See Also

DrawSelectionRect method

Images

SelectionRectangle property

ToolPaletteHidden Event

This event occurs immediately after the Annotation Tool Palette has been hidden.

Applies To

Image Edit control

Syntax

Sub *object*_**ToolPaletteHidden**(*Left,Top*)

Arguments

Parameter	Data Type	Setting
Left	Long	The horizontal position of the Annotation Tool Palette in pixels when it was closed, relative to the upper-left corner of the screen
Top	Long	The vertical position of the Annotation Tool Palette in pixels when it was closed, relative to the upper-left corner of the screen

Remarks

The Annotation Tool Palette can be hidden by invoking the **HideAnnotationToolPalette** method or by clicking on the Close icon on the Annotation Tool Palette caption bar.

See Also

Annotations

HideAnnotationToolPalette method

ToolTip Event

This event occurs immediately after a tool tip has been displayed on the Annotation Tool Palette.

Applies To

Image Edit control

Syntax

Sub *object_***ToolTip**(*Index*)

Argument

Parameter	Data Type	Setting
Index	Integer (enumerated)	Index value of the tool tip displayed, which corresponds to the Tool ID settings

Remarks

The **ShowAnnotationToolPalette** method displays the Annotation Tool Palette.   The **SelectTool** method selects the annotation tool in the Annotation Tool Palette.

See Also

Annotations

SelectTool method

ShowAnnotationToolPalette method

ToolSelected event

ToolSelected Event

This event occurs immediately after the end user selects a tool from the Annotation Tool Palette.

Applies To

Image Edit control

Syntax

Sub *object*_**ToolSelected**(*ToolID*)

Argument

Parameter	Data Type	Setting
ToolID	Integer (enumerated)	Identifier of the tool, which corresponds to the Tool ID settings

Remarks

The **ShowAnnotationToolPalette** method does not have to be invoked to trigger this event.   The **SelectTool** method can be used to select the annotation tool in the Annotation Tool Palette.

See Also

Annotations

SelectTool method

ShowAnnotationToolPalette method

ToolTip event

PageType Settings

The following list shows the valid PageType settings:

Setting	Description
1	Black-and-white
2	Gray4
3	Gray8
4	Palettized4 (not available as a parameter for the **ConvertPageType** method unless the image presently has a 4-bit palette)
5	Palettized8
6	RGB24
7	BGR24

PageType Settings

The valid PageType settings for the **SaveAs** method depend on the FileType setting entered.   The following list shows the valid PageType settings:

Setting	Description	Prerequisite FileType
1	Black-and-white	1 -- TIFF, 2 -- AWD, or 3 -- BMP
2	Gray4	1 -- TIFF
3	Gray8	1 -- TIFF
4	Palettized4	1 -- TIFF or 3 -- BMP
5	Palettized8	1 -- TIFF or 3 -- BMP
6	RGB24	1 -- TIFF
7	BGR24	3 -- BMP

To use a PageType setting of 4 (Palettized4), the displayed image must already be a Palettized4 image.

Annotation Type Settings

The following list shows the valid AnnotationType settings:

Setting	Description
0	None
1	Straight Line
2	Freehand Line
3	Hollow Rectangle
4	Filled Rectangle
5	Image Embedded
6	Image Reference
7	Text
8	Text Stamp
9	Text From File
10	Attach-a-Note
11	Select Annotations

FileType Settings

The following list shows the valid FileType settings:

Setting	Description
1	TIFF
2	AWD
3	BMP
4	PCX
5	DCX
6	JPG

Displayed file types of DCX, PCX, or JPG, must be saved using a FileType setting of 1, 2, or 3.

CompressionType Settings

The following list shows the valid CompressionType settings:

Setting	Description
1	No compression
2	Group3(1D)
3	Group3(Modified Huffman)
4	PackBits
5	Group4(2D)
6	JPEG

AWD and BMP files must be saved using a CompressionType setting of 1.

JPG files must be saved using a CompressionType setting of 6.

CompressionInfo Settings

The following list shows the valid CompressionInfo settings (bit-wise values):

Setting	Description
0	No compression -- If CompressionType is set to 1 (no compression)
	Default compression -- If CompressionType is set to a value other than 1
1	EOL
2	Packed Lines
4	Prefixed EOL
8	Compressed LTR
16	Expand LTR
32	Negate
64	High Compression/High Quality (JPEG compression only)
128	High Compression/Medium Quality (JPEG compression only)
256	High Compression/Low Quality (JPEG compression only)
512	Medium Compression/High Quality (JPEG compression only)
1024	Medium Compression/Medium Quality (JPEG compression only)
2048	Medium Compression/Low Quality (JPEG compression only)
4096	Low Compression/High Quality (JPEG compression only)
8192	Low Compression/Medium Quality (JPEG compression only)
16384	Low Compression/Low Quality (JPEG compression only)

AWD, BMP, and other uncompressed files must be saved using a CompressionInfo setting of 0.

Tool ID Settings

The following list shows the valid Tool ID settings:

Parameter	Data Type	Setting	Description
ToolID	Integer (enumerated)	0	No Tool
		1	Select Annotations
		2	Freehand Line
		3	Highlighter
		4	Straight Line
		5	Hollow Rectangle
		6	Filled Rectangle
		7	Text
		8	Attach-a-Note
		9	Text From File
		10	Rubber Stamp

Annotation Tool Palette

Click on the components of the tool palette to see an explanation of each one.

Select Annotations

The only tool that does not draw an annotation, the Select Annotations tool is used to select single or multiple annotation marks.   You can select annotation marks by clicking on them, or by dragging a selection rectangle around them.

Highlighter

The Highlighter tool is used to draw a transparent, filled rectangle for highlighting emphasis.

Hollow Rectangle

The Hollow Rectangle tool is used to place a border around areas of an image for emphasis.

Text

The Text tool is used to enter text directly on an image.

Text From File

The Text From File tool is used to enter text from a file onto an image.

Freehand Line

The Freehand Line tool is used to draw a freehand line on a section of text or a portion of an image for emphasis.

Straight Line

The Straight Line tool is used to underline text, demarcate a section of a page, or draw callout lines.

Filled Rectangle

The Filled Rectangle tool is used to create a color background for text annotations or to hide a portion of an image.

Attach-a-Note

The Attach-a-Note tool is used to enter text into a background rectangle on an image.

Rubber Stamp

The Rubber Stamp tool is used to place an image or text stamp onto an image.

Wang Image Admin Control

<u>Properties</u>   <u>Methods</u>

Wang's Image Admin control provides a method to administer image files that reside locally, or on a local area network (LAN). File and Print common dialog boxes support specific image file options as well as general directory and file functions.

General functions you can perform include creating, deleting, renaming, and moving directories and files, retrieving image file attributes, and setting image file attribute defaults.

Image-specific functions include inserting, appending, and replacing pages in a multi-page image file.

See Also

<u>Imaging Concepts</u>

Properties (Admin control)

CancelError	HelpContextId	PageNumber
CompressionInfo	HelpFile	PageType
CompressionType	HelpKey	PrintAnnotations
DefaultExt	Image	PrintEndPage
DialogTitle	ImageHeight	PrintNumCopies
FileType	ImageResolutionX	PrintOutputFormat
Filter	ImageResolutionY	PrintRangeOption
FilterIndex	ImageWidth	PrintStartPage
Flags	InitDir	PrintToFile
HelpCommand	PageCount	StatusCode

CancelError Property

Indicates or sets if an error will be returned when a user chooses Cancel from either the File or Print dialog boxes.

Applies To

Image Admin control

Syntax

object.**CancelError**[= {*True*|*False*}]

Data Type

Boolean

Setting	Description
True (default)	Causes an error to be returned when the user chooses Cancel from either the File or Print dialog boxes.
False	No error is returned when the user chooses Cancel.

Remarks

When a user chooses Cancel from the File or Print dialog boxes, and this property is set to True, an error is returned by the **ShowFileDialog** or **ShowPrintDialog** methods, respectively.

See Also

ShowFileDialog method

ShowPrintDialog method

CompressionInfo Property

Returns the bitwise compression options set for the current image.

Applies To

Image Admin control

Syntax

*object.***CompressionInfo**[=*value*]

Data Type

Long

Setting	Description
Note:	*Values 1 to 32 are not used with JPEG compression.*
1	EOL -- include or expect standard end of line bit sequences
2	Packed lines -- data is not byte aligned
4	Prefixed EOLs -- include or expect prefixed end of line bit sequences
8	Compressed bit order, left to right
16	Expanded bit order, left to right
32	Negate -- indicates data is stored with black bits equal to 0 (zero)
Note:	*Values 64 to 16384 pertain only to JPEG compression.*
64	High compression, high quality
128	High compression, medium quality
256	High compression, low quality
512	Medium compression, high quality
1024	Medium compression, medium quality
2048	Medium compression, low quality
4096	Low compression, high quality
8192	Low compression, medium quality
16384	Low compression, low quality

Remarks

The value returned by this property is for the image page specified by the **PageNumber** property. If no image is specified by PageNumber, the value returned is 0 (zero).

This property is read-only.

See Also

Image property

Compression

PageNumber property

CompressionType property

CompressionType Property

Returns the compression type of the current image.

Applies To

Image Admin control

Syntax

*object.***CompressionType**[=*value*]

Data Type

Integer (enumerated)

Setting	Description
0	Unknown
1	No compression
2	Group 3 1D FAX
3	Group 3 Modified Huffman
4	Packed Bits
5	Group 4 2D FAX
6	JPEG

Remarks

The value returned by this property is for the image page specified by the **PageNumber** property. If the **Image** property is empty, the value returned is 0 (zero).

This property is read-only.

See Also

Image property

Compression

PageNumber property

CompressionInfo property

DefaultExt Property

Returns or sets the default file extension in the ShowFileDialog dialog box.

Applies To

Image Admin control

Syntax

*object.***DefaultExt**[=*extension*]

Data Type

String

Remarks

The extension of the currently selected filter overrides the default extension.

The default extension is used only when there is no extension associated with the filter (for example, 'All files *.*').

The first three characters of the string you specify are used as the file extension when the user does not specify a file extension.

See Also

ShowFileDialog method

DialogTitle Property

Returns or sets the title used in the common dialog boxes.

Applies To

Image Admin control

Syntax

*object.***DialogTitle**[*=title*]

Data Type

String

Remarks

This property is used in conjunction with the **ShowFileDialog** method.

If a string is not specified, the default is used -- either Open or Save As.

See Also

ShowFileDialog method

FileType Property

Returns the file format of the image file.

Applies To

Image Admin control

Syntax

*object.***FileType**[*=value*]

Data Type

Integer (enumerated)

Property settings are expressed using the following values:

Setting	Description
0	Unknown
1	TIFF
2	AWD
3	Bitmap (BMP)
4	PCX
5	DCX
6	JPEG

Remarks

The file format is determined by the **Image** property.

This property is read-only.

See Also

File types

Image property

Filter Property

Returns or sets the list of file filters displayed.

Applies To

Image Admin control

Syntax

*object.***Filter**[*=filter*]

Data Type

String

Remarks

Filters are displayed in the common dialog box created by the **ShowFileDialog** method.

Each filter type consists of two strings separated by, and ending with, a pipe symbol (|). The first string contains a description of the filter, such as 'Tiff files'. The second string contains the file extension, such as '*.tif'. Be sure not to include spaces immediately before and after the pipe symbol, or they become part of the string.   For example:

All image files|*.tif;*.awd;*.bmp|TIFF files|*.tif|AWD files|*.awd|BMP files|*.bmp|

If no filter type is provided, the system supplies a default filter.

See Also

ShowFileDialog method

FilterIndex Property

Specifies which of the file filters is highlighted in the file common dialog box when the dialog box first opens.

Applies To

Image Admin control

Syntax

*object.***FilterIndex**[*=index*]

Data Type

Long

Remarks

If no filters are specified by the **Filter** property and the **FilterIndex** property is set to 0 (zero), then a default filter will be highlighted. If a different filter is selected, the index still remains set to 0.

If a value was set for the Filter property, and a different filter is selected, the index will be updated to reflect the selected filter.

The default value is 0 (zero).

See Also

Filter property

ShowFileDialog method

Flags Property

Sets initialization flags for dialog boxes created by the ShowFileDialog and ShowPrintDialog methods.

Applies To

Image Admin control

Syntax

*object.***Flags**[=*value*]

Data Type

Long

Values for ShowFileDialog

Value	Description
&H400&	Indicates that the extension of the returned filename is different from the extension specified by the DefaultExt property. This flag is not set if the DefaultExt property is empty, if the extensions match, or if the file has no extension. This flag value can be checked upon returning from the dialog box.
&H4&	Hides the Read-Only check box.
&H8&	Forces the dialog box to set the current directory to what it was when the dialog box was invoked.
&H8000&	Specifies that the returned file will not have the Read-Only attribute set and will not be in a write-protected directory.
&H100&	Specifies that the common dialog box allows invalid characters in the returned filename.
&H1&	Causes the Read Only check box to be initially checked when the dialog box is created. This flag also indicates the state of the Read Only check box when the dialog box is closed.
&H10&	Causes the dialog box to display the Help button. If this option is specified, the parameter hParentWnd in the ShowFileDialog method must be set to null.

Values for ShowPrintDialog

Value	Description
&H10&	Sets or returns the state of the Collate check box.
&H80000&	Disables the Print to File check box.
&H100000&	The Print to File check box is not displayed.
&H80&	Prevents a warning message from being displayed when there is no default printer.
&H400&	Causes the ShowPrintDialog method to return

the system default printer. The ShowPrintDialog method does this without displaying the dialog box.

&H40000&   If a printer driver does not support multiple copies, setting this flag disables the copies' edit control. If a driver does support multiple copies, setting this flag indicates that the dialog box should store the requested number of copies in the Copies property.

Remarks

The flags are bitwise and can be OR'd together.

These flags are defined in the Visual Basic CONSTANT.TXT file.

The Help button in the dialog box is not shown unless a Help file is specified by the **HelpFile** property.

See Also

HelpFile property
ShowFileDialog method

HelpCommand Property

Returns or sets the type of Help requested.

Applies To

Image Admin control

Syntax

*object.***HelpCommand**[*=value*]

Data Type

Integer

Setting	Description
&H1&	Displays Help for a particular context. When using this setting, you must also specify a context using the HelpContext property.
&H3&	Displays the index of the specified Help file. An application should use this value only for a Help file with a single index.
&H4&	Displays Help for using the Help application itself.
&H5&	Sets the context specified by the HelpContext property as the current index for the Help file specified by the HelpFile property. This index remains current until the user accesses a different Help file. Use this value only for Help files with more than one index.
&H101&	Displays Help for a particular keyword. When using this setting, you must also specify a keyword using the HelpKey property.
&H105&	Brings up a search box in Help, with the key specified by the HelpKey property.

Remarks

Values for the HelpCommand property constants are defined in the VisualBasic CONSTANT.TXT file.

See Also

HelpContextId property
HelpKey property
HelpFile property
ShowFileDialog method

HelpContextId Property

Returns or sets a Help context Id.

Applies To

Image Admin control

Syntax

*object.***HelpContextId**[=*Value*]

Data Type

Integer

Remarks

Context Ids are assigned to objects to provide context-sensitive Help to your application.

See Also

[HelpCommand property](#)
[HelpKey property](#)
[HelpFile property](#)
[ShowFileDialog method](#)

HelpKey Property

Returns or sets the keyword that identifies the requested Help topic.

Applies To

Image Admin control

Syntax

*object.***HelpKey**[*=keyword*]

Data Type

String

Remarks

Use this property with the **HelpCommand** property to specify a Help topic to be displayed.

See Also

HelpContextId property
HelpCommand property
HelpFile property
ShowFileDialog method

HelpFile Property

Returns or sets the name of a Help file to be used with the application.

Applies To

Image Admin control

Syntax

*object.***HelpFile**[*=filename*]

Data Type

String

See Also

HelpContextId property
HelpCommand property
HelpKey property
ShowFileDialog method

Image Property

Returns or sets the filename of the current image.

Applies To

Image Admin control

Syntax

*object.***Image**[*=filename*]

Data Type

String

Remarks

When a new image is specified, the PageNumber property is set to 1.

This property is used by the following Image Admin control methods:

Append	DeletePages	Insert	VerifyImage
Replace	ShowFileDialog	ShowPrintDialog	

When the Image property changes, the following Admin control properties are updated:

CompressionInfo	ImageResolutionX	PageNumber
CompressionType	ImageResolutionY	PageType
FileType	ImageWidth	
ImageHeight	PageCount	

See Also

Images

ImageHeight Property

Returns the height of the current page.

Applies To

Image Admin control

Syntax

*object.***ImageHeight**[*=value*]

Data Type

Long

Remarks

The page is part of the current image file, as specified by the **Image** property.

Height is indicated in pixels.

This property is read-only.

See Also

PageNumber property
ImageWidth property
Image property

ImageResolutionX Property

Returns the horizontal resolution of the current page.

Applies To

Image Admin control

Syntax

*object.***ImageResolutionX**[=*value*]

Data Type

Long

Remarks

The page is part of the current image file, as specified by the **Image** property.

Resolution is in dots-per-inch (dpi).

This property is read-only.

See Also

ImageResolutionY property
Image property
PageNumber property

ImageResolutionY Property

Returns the vertical resolution of the current page.

Applies To

Image Admin control

Syntax

*object.***ImageResolutionY**[*=value*]

Data Type

Long

Remarks

The page is part of the current image file, as specified by the **Image** property.

Resolution is in dots-per-inch (dpi).

This property is read-only.

See Also

ImageResolutionX property
Image property
PageNumber property

ImageWidth Property

Returns the width of the current page.

Applies To

Image Admin control

Syntax

*object.***ImageWidth**[=*value*]

Data Type

Long

Remarks

The page is part of the current image file, as specified by the **Image** property.

Width is indicated in pixels.

This property is read-only.

See Also

ImageHeight property

Image property

PageNumber property

InitDir Property

Returns or sets the initial directory displayed in the ShowFileDialog dialog box.

Applies To

Image Admin control

Syntax

*object.***InitDir**[*=directoryname*]

Data Type

String

Remarks

If the specified directory is invalid, the current directory appears on the dialog box.

See Also

ShowFileDialog method

PageCount Property

Returns the number of pages in the image file.

Applies To

Image Admin control

Syntax

*object.***PageCount**[*=value*]

Data Type

Long

Remarks

The image file is specified by the **Image** property.

The value is 0 (zero) if the image specified does not exist.

This property is read-only.

See Also

Image page
Image property

PageNumber Property

Returns or sets a page number in the current image file.

Applies To

Image Admin control

Syntax

*object.***PageNumber**

Data Type

Long

Remarks

This property is available only at run-time (read and write).

The image file is specified by the **Image** property.

If the image file is not multi-page, this value must be 1.

See Also

Image page

Image property

PageType Property

Returns the image type of the specified page.

Applies To

Image Admin control

Syntax

*object.***PageType**[=*value*]

Data Type

Integer (enumerated)

Setting	Description
0	Unknown
1	Black and white
2	4-bit gray scale
3	8-bit gray scale
4	4-bit palettized
5	8-bit palettized
6	24-bit RGB
7	24-bit BGR

Remarks

The Image file is specified by the **Image** property; the page is specified by the **PageNumber** property.

If the Image property is empty, the value is 0 (zero).

This property is read-only.

See Also

Image types

Image property

PageNumber property

PrintAnnotations Property

Indicates if the Print Annotations check box in the Print dialog box is checked.

Applies To

Image Admin control

Syntax

*object.***PrintAnnotations**[={*True|False*}]

Data Type

Boolean

Setting	Description
True (default)	The check box is checked, and annotations are to be printed.
False	The check box is not checked and annotations are not to be printed.

Remarks

This property sets the state of the check box that is displayed when the dialog box is first opened. The user can accept or change this setting.

The Print dialog box is created by the **ShowPrintDialog** method.

See Also

Annotations

ShowPrintDialog method

PrintEndPage Property

Returns or sets the number of the last page to be printed.

Applies To

Image Admin control

Syntax

*object.***PrintEndPage**

Data Type

Long

Remarks

The value set by this property is displayed when the Print dialog box is first opened. The user can then accept or change that value.

The default value is the value specified in the PageCount property.

The Print dialog box is created by the **ShowPrintDialog** method.

This property is available only at run-time (read and write).

See Also

Image page

PrintStartPage property

ShowPrintDialog method

PrintNumCopies Property

Returns or sets the default number of copies for the Print Dialog box.

Applies To

Image Admin control

Syntax

*object.***PrintNumCopies**[*=value*]

Data Type

Long

Remarks

The Print dialog box is created by the **ShowPrintDialog** method.

The value set by this property is displayed when the Print dialog box is first opened. The user can then accept or change that value.

The value entered by the user is returned by the ShowPrintDialog method.

See Also

ShowPrintDialog method

PrintOutputFormat Property

Returns or sets the format of the printer output.

Applies To

Image Admin control

Syntax

*object.***PrintOutputFormat**[*=value*]

Data Type

Integer (enumerated)

Setting	Description
0	Pixel-to-pixel
1	Actual size
2 (default)	Fit to page

Remarks

The Print dialog box is created by the **ShowPrintDialog** method.

The value set by this property is displayed when the Print dialog box is first opened. The user can then accept or change that value.

See Also

ShowPrintDialog method

PrintRangeOption Property

Returns or sets the print range option for the Print dialog box.

Applies To

Image Admin control

Syntax

*object.***PrintRangeOption**[=*value*]

Data Type

Integer (enumerated)

Setting	Description
0 (default)	All pages
1	Range of pages
2	Current page

Remarks

The value set by this property is displayed when the Print dialog box is first opened. The user can then accept or change that value.

If a value of 1 is set, the start and end pages are provided by the **PrintStartPage** and **PrintEndPage** properties.

The Print dialog box is created by the **ShowPrintDialog** method.

See Also

Image page

PrintStartPage property

PrintEndPage property

PrintStartPage Property

Returns or sets the number of the first page to be printed for the Print dialog box.

Applies To

Image Admin control

Syntax

*object.***PrintStartPage**

Data Type

Long

Remarks

The value set by this property is displayed when the Print dialog box is first opened. The user can then accept or change that value.

The Print dialog box is created by the **ShowPrintDialog** method.

The default value is 1.

This property is available only at run-time (read and write).

See Also

Image page

PrintEndPage property

ShowPrintDialog method

PrintToFile Property

Indicates or sets if the Print to File check box in the Print dialog box is checked.

Applies To

Image Admin control

Syntax

*object.***PrintToFile**[={*True|False*}]

Data Type

Boolean

Setting	Description
True	The check box is checked, and data is to be printed to a file.
False (default)	The check box is not checked, and data is not to be sent to a file.

Remarks

The Print dialog box is created by the **ShowPrintDialog** method.

This property sets the state of the check box that is displayed when the dialog box is first opened. The user can accept or change this setting.

The state of the check box is returned by the ShowPrintDialog method.

Methods (Admin control)

AboutBox

Append

CreateDirectory

Delete

DeletePages

Insert

GetSysCompressionInfo

GetSysCompressionType

GetSysFileType

GetUniqueName

Replace

SetSystemFileAttributes

ShowFileDialog

ShowPrintDialog

VerifyImage

Append Method

Appends one or more pages to the current image.

Applies To

Image Admin control

Syntax

*object.***Append** *Source, SourcePage, NumPages* [,*CompressionType, CompressionInfo*]

Arguments

Parameter	Data Type	Description
Source	String	Specifies the image file that contains the page(s) to be appended.
SourcePage	Long	Specifies the first page in the source image file to be appended.
NumPages	Long	Specifies the number of pages from the source image file to append to the current image file.
CompressionType	Integer	(Optional) Specifies the compression type to apply to the appended pages. If a type is not specified, the compression type of the source page(s) is used.
CompressionInfo	Long	(Optional) Specifies which compression options to apply to the appended pages. If options are not specified, the compression options of the source page(s) are used.

Remarks

The current image file is specified by the **Image** property.

The current image, if one exists, must be in either AWD or TIFF format, as these formats support multiple pages.   If the image pages added are of another type, they will be converted to the file type format used by the current image.

You can append a single BMP file to a new file. If you do, the only compression type allowed is None, with no options.

If the CompressionType and CompressionInfo values specified are not compatible with a source page's page type (color), the source file's compression type and compression information are used in place of those arguments.

If JPEG compression type is specified, and the compression argument is 0 (zero), a default JPEG value is used. The default value is 4096 (low compression, high quality).

If the destination file is in AWD format, only black and white files without annotations may be appended.

The only allowable compression type for AWD is None, with no options.

See Also

Compression

CompressionType property

CompressionInfo property

CreateDirectory Method

Creates a user-defined directory.

Applies To

 Image Admin control

Syntax

 *object.***CreateDirectory** *Path*

Arguments

Parameter	Data Type	Description
Path	String	Specifies the path to be created.

Remarks

 If the Path contains more than one subdirectory that does not exist, an error is returned.

Delete Method

Deletes a user-specified object.

Applies To

Image Admin control

Syntax

*object.***Delete** *Object*

Arguments

Parameter	Data Type	Description
Object	String	Specifies the object to be deleted.

Remarks

The Object specified can be a filename or a path.

A file cannot be deleted if it is being used by any Wang Image application or OCX control.

DeletePages Method

Deletes a user-specified range of pages from the image file.

Applies To

Image Admin control

Syntax

*object.***DeletePages** *StartPage, NumPages*

Arguments

Parameter	Data Type	Description
StartPage	Long	First page to be deleted.
NumPages	Long	Number of pages to be deleted, including StartPage.

Remarks

The image file is specified by the **Image** property.

Pages cannot be deleted if they are being used by any Wang Image application or OCX control.

If StartPage is within the image, and NumPages is greater than the number of pages in the image, all the pages after StartPage are deleted.

See Also

Image page

Insert Method

Inserts one or more user-specified pages into the current image file.

Applies To

Image Admin control

Syntax

object.**Insert** *Source, SourcePage, DestinationPage, NumPages* [,*CompressionType, CompressionInfo*]

Arguments

Parameter	Data Type	Description
Source	String	Specifies the image file that contains the page(s) to be inserted.
SourcePage	Long	Specifies which page in the source image file will be inserted.
DestinationPage	Long	Specifies the page in the current image file the selected pages will be inserted before.
NumPages	Long	Specifies the number of pages from the source image file to insert into the current image file.
CompressionType	Integer	(Optional) Specifies the compression type to apply to the inserted pages. If a type is not specified, the compression type of the source image is used.
CompressionInfo	Long	(Optional) Specifies which compression options to apply to the inserted pages. If options are not specified, the compression options of the source image are used.

Remarks

The current image is specified by the **Image** property.

The current image must be in either AWD or TIFF format, because these formats support multiple pages. If the image pages added are of another type, they will be converted to the file type used by the current image.

You can insert a single BMP page into a new image.

If the CompressionType and CompressionInfo values specified are not compatible with a source page's page type, the source file's compression type and compression information is used in place of the arguments.

If JPEG compression is specified, and the compression argument is 0 (zero), a default JPEG value is used. The default value is 4096 (low compression, high quality).

If the destination file is in AWD format, only black and white files without annotations may be inserted.

The only allowable compression type for AWD is None, with no options.

See Also

[Compression](#)

[CompressionType property](#)

[CompressionInfo property](#)

GetSysCompressionInfo Method

Retrieves the system default compression information for a given page type.

Applies To

Image Admin control

Syntax

*object.***GetSysCompressionInfo**(*PageType*)

Arguments

Parameter	Data Type	Description
PageType	Integer	Specifies a page type.

Setting	Description
0	Unknown
1	Black and white
2	4-bit gray scale
3	8-bit gray scale
4	4-bit palettized
5	8-bit palettized
6	24-bit RGB
7	24-bit BGR

Return Value

Long

Returns the following bitwise compression information:

Setting	Description
1	EOL -- include or expect standard end of line bit sequences
2	Packed lines -- data is not byte aligned
4	Prefixed EOLs -- include or expect prefixed end of line bit sequences
8	Compressed bit order, left to right
16	Expanded bit order, left to right
32	Negate -- reverses black and white on expansion
64	High compression, high quality
128	High compression, medium quality
256	High compression, low quality
512	Medium compression, high quality

1024	Medium compression, medium quality
2048	Medium compression, low quality
4096	Low compression, high quality
8192	Low compression, medium quality
16384	Low compression, low quality

See Also

Compression

Image types

SetSystemFileAttributes method

GetSysCompressionType Method

Retrieves the system default compression type for a given page type.

Applies To

Image Admin control

Syntax

*object.***GetSystemCompressionType**(*PageType*)

Arguments

Parameter	Data Type	Description
PageType	Integer	Specifies a page type.

Setting	Description
0	Unknown
1	Black and white
2	4-bit gray scale
3	8-bit gray scale
4	4-bit palettized
5	8-bit palettized
6	24-bit RGB
7	24-bit BGR

Return Value

Integer (enumerated)

Returns one of the following compression types:

Setting	Description
0	Unknown
1	No compression
2	Group 3 1D FAX
3	Group 3 Modified Huffman
4	Packed Bits
5	Group 4 2D FAX
6	JPEG

See Also

GetSystemFileAttributes method

Image types

<u>Compression types</u>

GetSysFileType Method

Retrieves the system default file type for a given page type.

Applies To

Image Admin control

Syntax

*object.***GetSysFileType**(*PageType*)

Arguments

Parameter	Data Type	Description
Page Type	Integer	Specifies a page type.

Setting	Description
0	Unknown
1	Black and white
2	4-bit gray scale
3	8-bit gray scale
4	4-bit palettized
5	8-bit palettized
6	24-bit RGB
7	24-bit BGR

Return Value

Integer (enumerated)

One of the following file types is returned:

Setting	Description
0	Unknown
1	TIFF
2	AWD
3	Bitmap (BMP)

See Also

SetSystemFileAttributes method
Image types
File types

GetUniqueName Method

Generates a unique filename.

Applies To

Image Admin control

Syntax

*object.***GetUniqueName**(*Path, Template, Extension*)

Arguments

Parameter	Data Type	Description
Path	String	The name of the existing path where the filename will be generated.
Template	String	The initial characters of the filename. Up to 4 characters can be used.
Extension	String	The extension to be used with the generated filename. Up to 3 characters can be used.

Return Value

A string that contains a fully-qualified filename.

Remarks

The filename is based on user-supplied path, template, and file extension information.

Replace Method

Replaces one or more user-specified pages into the current image file.

Applies To

Image Admin control

Syntax

*object.***Replace** *Source, SourcePage, DestinationPage, NumPages* [,*CompressionType*, *CompressionInfo*]

Arguments

Parameter	Data Type	Description
Source	String	Specifies the image file that contains the page(s) to be inserted.
SourcePage	Long	Specifies the first page in the source image file that will be inserted.
DestinationPage	Long	Specifies the first page in the current image file that will be replaced.
NumPages	Long	Specifies the number of pages in the current image file to replace with pages from the source image file.
CompressionType	Integer	(Optional) Specifies the compression type to apply to the replaced pages. If a type is not specified, the compression type of the source image is used.
CompressionInfo	Long	(Optional) Specifies which compression options to apply to the replaced pages. If options are not specified, the compression options of the source page are used.

Remarks

The current image is specified by the **Image** property.

The replaced pages are converted to the file type of the current image (TIFF, AWD, or BMP).

If you are replacing the single page in a BMP file, the only compression type allowed is None, with no options.

If the CompressionType and CompressionInfo values specified are not compatible with a source page's page type, the source file's compression type and compression information is used in place of the arguments.

If JPEG compression is specified, and the compression argument is 0 (zero), a default JPEG value is used. The default value is 4096 (low compression, high quality).

If the destination file is in AWD format, only black and white files without annotations may be

replaced.

The only allowable compression type for AWD is none, with no options.

See Also

[Compression](#)
[CompressionInfo property](#)
[CompressionType property](#)

SetSystemFileAttributes Method

Allows the user to set how image files will be saved for a given type.

Applies To

Image Admin control

Syntax

*object.***SetSystemFileAttributes** *PageType, FileType, CompressionType, CompressionInfo*

Arguments

Parameter	Data Type	Description
PageType	Integer	Specifies a page type value.
FileType	Integer	Specifies a file type value.
CompressionType	Integer	Specifies compression information.
CompressionInfo	Long	Specifies compression information.

Remarks

These settings will be used when scanning files with the Image Scan control.

Defaults are set for a classification of the page type into one of three categories:
- Black and White
- Gray scale
- Color

If a compression option of 0 (zero) is specified for a JPEG compression type, then a default value will be used (4096 -- low compression, high quality).

See Also

File types

Image types

Compression

GetSysCompressionType

GetSysFileType

GetSysCompressionInfo

ShowFileDialog Method

Allows a user to select a file specification from the file system displayed in an Open or Save As dialog box.

Applies To

Image Admin control

Syntax

*object.***ShowFileDialog** *DialogOption*[*,hParentWnd*]

Arguments

Parameter	Data Type	Description
DialogOption	Integer	Indicates what type of file dialog box will be displayed:
		0= Open or Select 1= Save As
hParentWnd	hWnd	(Optional) Assigns a parent window handle to the displayed dialog box.

Remarks

The file specification selected by the user from the dialog box is set in the **Image** property when the method returns the value.

If a help file is not specified in the **HelpFile** property, a Help button will not be shown. This is true even if the show help flag is set in the **Flags** property.

Dialog boxes are modal.

See Also

HelpFile property
HelpContextId property
HelpCommand property
HelpKey property
Flags property
Image property

ShowPrintDialog Method

Presents a printer dialog box from which the user sets various print options.

Applies To

Image Admin control

Syntax

*object.***ShowPrintDialog** [*hParentWnd*]

Arguments

Parameter	Data Type	Description
hParentWnd	hWnd	(Optional) Assigns a parent window handle to the displayed dialog box.

Remarks

The image property must be set to an existing image or an error is returned.

This function returns any parameters set by the user to the application via the corresponding properties.

Dialog boxes are modal.

See Also

PrintAnnotations property
PrintEndPage property
PrintNumCopies property
PrintOutputFormat property
PrintRangeOption property
PrintStartPage property
PrintToFile property
Flags property
Image property

VerifyImage Method

Checks to see if the image specified in the Image property exists, and verifies level of access.

Applies To

Image Admin control

Syntax

*object.***VerifyImage**(*Option*)

Arguments

Parameter	Data Type	Description
Option	Long	Specifies the level of access to verify. 0 = Verify existence 1 = Verify read access 2 = Verify write access 3 = Verify read/write access

Return Value

Boolean

Setting	Description
True	The file exists and has the level of access specified by the Option argument.
False	The file does not exist, or does not have the level of access specified in the Option argument.

File Type argument

Setting	Description
0	Unknown
1	TIFF
2	AWD
3	Bitmap (BMP)
4	PCX
5	DCX
6	JPEG

PageType argument

Setting	Description
0	Unknown
1	Black and white
2	4-bit gray scale
3	8-bit gray scale
4	4-bit palettized
5	8-bit palettized
6	24-bit RGB
7	24-bit BGR

CompressionType argument

Setting	Description
0	Unknown
1	No compression
2	Group 3 1D FAX
3	Group 3 Modified Huffman
4	Packed Bits
5	Group 4 2D FAX
6	JPEG

CompressionInfo argument

Setting	Description
1	EOL -- include or expect standard end of line bit sequences
2	Packed lines -- data is not byte aligned
4	Prefixed EOLs -- include or expect prefixed end of line bit sequences
8	Compressed bit order, left to right
16	Expanded bit order, left to right
32	Negate -- reverses black and white on expansion
64	High compression, high quality
128	High compression, medium quality
256	High compression, low quality
512	Medium compression, high quality
1024	Medium compression, medium quality
2048	Medium compression, low quality
4096	Low compression, high quality
8192	Low compression, medium quality
16384	Low compression, low quality

Wang Image Scan Control

Properties　　Methods　　Events

Wang's Image Scan control adds scanning capabilities to applications that support 32-bit OLE controls. The Scan control is invisible and can work independently of other controls. To display images while they are being scanned, you can link the Scan control to the Wang Image Edit control.

The Image Scan control supports TWAIN-compatible scanners. For a list of scanners that have been certified with Imaging software, refer to the README.DOC file included with your Wang software or contact the manufacturer of your scanner.

Your scanner will require a TWAIN data source (usually supplied by the manufacturer) and version 1.6 of the TWAIN DLLs (TWAIN.DLL, TWAIN_32.DLL, TWUNK_16.EXE, and TWUNK_32.EXE). Version 1.6 of the TWAIN DLLs are included with the Imaging kit, and can be installed by checking the 'TWAIN 1.6' box during setup.

Notes: After installing version 1.6 of the TWAIN DLLs, you may experience problems running applications that are not compatible with version 1.6.

If you install a new TWAIN data source after installing Imaging, you must then reinstall Imaging, checking both the TWAIN 1.6 box and the Imaging box.

Some scanners may have been sold with data sources that are not compatible with your current operating system. Please contact the manufacturer of your scanner for the latest data sources and compatibility information.

See Also

Imaging Concepts

Properties (Scan control)

CompressionInfo   PageOption

CompressionType   PageType

DestImageControl   ScanTo

FileType   Scroll

Image   StatusCode

MultiPage   StopScanBox

Page   Zoom

PageCount

CompressionInfo Property

Sets the compression options to use when scanning.

Applies To

Image Scan control

Syntax

*object.***CompressionInfo**[=*value*]

Data Type

Long

Setting	Description
Note: Settings 1 to 32 cannot be used with JPEG compression.	
1	EOL -- include or expect standard end of line bit sequences
2	Packed lines -- data is not byte aligned
4	Prefixed EOLs -- include or expect prefixed end of line bit sequences
8	Compressed bit order, left to right
16 (default)	Expanded bit order, left to right
32	Negate -- reverses black and white on expansion
Note: The following settings pertain only to JPEG compression.	
64	High compression, high quality
128	High compression, medium quality
256	High compression, low quality
512	Medium compression, high quality
1024	Medium compression, medium quality
2048	Medium compression, low quality
4096	Low compression, high quality
8192	Low compression, medium quality
16384	Low compression, low quality

Remarks

Using the JPEG compression method, the higher the compression ratio, the more degraded the image quality becomes. The amount of degradation that is acceptable varies with the image and the application. Experimenting with different settings is recommended.

Resolution is independent of the compression ratio. Optimal settings will vary, based on individual requirements.

Be careful when changing these values -- the compression type could be affected.

See Also

Compression Options

FileType property

PageType property

CompressionType property

Zoom property

CompressionType Property

Sets the compression type to use when scanning.

Applies To

Image Scan control

Syntax

*object.***CompressionType**[=*value*]

Data Type

Integer (enumerated)

Setting	Description
1	No compression
2	Group 3 1D FAX
3 (default)	Group 3 1D Modified Huffman
4	Packed Bits
5	Group 4 2D FAX
6	JPEG

Remarks

The value set for this property determines which settings are available in the **CompressionInfo** property.

See Also

Compression types

FileType property

PageType property

CompressionInfo property

Zoom property

DestImageControl Property

Returns or sets the name of the destination image / edit control.

Applies To

Image Scan control

Syntax

*object.***DestImageControl**[=*controlname*]

Data Type

String

Remarks

This property links the Image Scan control to a specified Wang ImageEdit control, letting you see the image that is scanned.

Use the name of the ImageEdit control to which you want to link.

See Also

ImageControl property (Wang ImageEdit control)

FileType Property

Returns or sets the type of image file that will be created.

Applies To

Image Scan control

Syntax

*object.***FileType**[*=value*]

Data Type

Integer (enumerated)

Setting	Description
1 (default)	TIFF
2	AWD
3	BMP

Remarks

Set to the file type you want to create.

See Also

File types
PageType property
CompressionType property
CompressionInfo property
Zoom property

Image Property

Returns or sets the name of the object to which you are scanning.

Applies To

Image Scan control

Syntax

*object.***Image**[=*filename*]

Data Type

String

Remarks

This property is not the same as, or linked to, the Image properties used by the Wang ImageEdit or Thumbnail controls.

You can point to a file template, and have image files generated and incremented as you scan to them. For example, if you use a template named 'img', you must specify a path, such as c:\scan\images\img. The template 'img' will generate files named img00000.xxx, img00001.xxx, img00002.xxx, and so on. This cannot be used if you Scan to display using the **DestImageControl** property in conjunction with the Wang Image Edit control.

See Also

Images

DestImageControl property

MultiPage Property

Returns or sets if one or more image pages will be scanned to an image file.

Applies To

Image Scan control

Syntax

object.**MultiPage**[= {*True|False*}]

Data Type

Boolean

Setting	Description
True	Scans multiple image pages to an image file.
False (default)	Scans a single image page to an image file.

Remarks

This property is generally used with scanners that have an automatic document feeder (ADF), and you are scanning multiple documents to a template.

See Also

Image page

Page Property

Returns or sets the starting page for a scanning session.

Applies To

Image Scan control

Syntax

*object.***Page**[*=pagenumber*]

Data Type

Long

Remarks

Set this property to the current page number.

To append to a file, set the page number to 0 (zero).

To insert one or more page, set the page number to the number of the page you are inserting before, less 1. For example, to insert before page 5, enter page number 4.

See Also

Image page

PageCount Property

Returns or sets the number of pages per image file.

Applies To

Image Scan control

Syntax

*object.***PageCount**[*=pagenumber*]

Data Type

Long

Remarks

This property is generally used with scanners that have an automatic document feeder (ADF), and you are scanning multiple documents to a template.

Set this property to the page count of fixed length files. You must also specify the path to the template (for example c:\scan\images\img, where the Template name is img).

If the **MultiPage** property is set to False, then PageCount images are scanned with the same number of image files.

See Also

Image page

MultiPage property

PageOption Property

Indicates or sets if a scanned page will be Appended or Inserted, or if a page will be overwritten with or without a prompt at scan time.

Applies To

Image Scan control

Syntax

*object.***PageOption**[*=value*]

Data Type

Integer (enumerated)

Setting	Description
0	Creates a new image file and adds image pages. An error will occur if the file exists. Pages can be added only if an automatic document feeder (ADF) is present.
1 (default)	Creates a new image file and adds image pages. If an image file exists, a prompt is displayed before overwriting. Pages can be added only if an automatic document feeder (ADF) is present.
2	Appends to an existing image file.
3	Inserts a page in the existing image file, before the current image page, as specified in the Page property. An error will occur if the file does not exist.
4	Overwrites an image file on an existing image page. If the image page does not exist, an error will occur.
5	Displays a prompt prior to overwriting an existing image page. If the image page does not exist, an error will occur.
6	Overwrites all pages in an image file.

Remarks

The Page property determines the current image page in a selected file.

If settings 0, 1, 2, or 6 are chosen, a new file is created if the file specified in the **Image** property does not exist.

See Also

Page property

Image property

PageType Property

Returns or sets the image type for the pages being saved after scanning.

Applies To

Image Scan control

Syntax

*object.***PageType**[=*value*]

Data Type

Integer (enumerated)

Setting	Description
1 (default)	Black and white
2	Gray scale, 4-bit
3	Gray scale, 8-bit
4	Palettized, 4-bit (BMP only)
5	Palettized, 8-bit (TIFF only)
6	RGB, 24-bit
7	BGR, 24-bit

Remarks

This property also determines the compression type and compression information required for saving the image file.

See Also

Image Types
FileType property
CompressionType property
CompressionInfo property
Zoom property

ScanTo Property

Returns or sets the destination of the image being scanned.

Applies To

Image Scan control

Syntax

*object.***ScanTo**[=*value*]

Data Type

Integer (enumerated)

Setting	Description
0 (default)	Display the scanned image.
1	Display the scanned image and write it to a file.
2	Write the scanned image to a file.
3	Write the scanned image to a file using a template, and display the image.
4	Write the scanned image to a file using a template.
5	Fax the scanned image.

Remarks

Images scanned to display must be saved in the Wang ImageEdit control.

Ensure that the ImageEdit control's **Image** property is set correctly.

See Also

Image property, Wang ImageEdit control

MultiPage property

PageCount property

Scroll Property

Sets or indicates if a displayed image will be scrolled while it is being scanned.

Applies To

Image Scan control

Syntax

object.**Scroll**[={*True|False*}]

Data Type

Boolean

Setting	Description
True (default)	Scanned image is scrolled.
False	Scanned image is not scrolled.

Remarks

Used in conjunction with the **ScanTo** property if ScanTo is set to a value of 0, 1, or 3.

See Also

ScanTo property

StopScanBox Property

Sets or indicates if a dialog box from which you can stop the scanning session will be displayed.

Applies To

Image Scan control

Syntax

object.**StopScanBox**[={*True*|*False*}]

Data Type

Boolean

Setting	Description
True	Displays the Stop Scan dialog box.
False (default)	Does not display the Stop Scan dialog box.

Zoom Property

Returns or sets the scale at which scanned pages are displayed.

Applies To

Image Scan control

Syntax

*object.***Zoom**[*=value*]

Data Type

Long

Remarks

Used in conjunction with the **ScanTo** property if ScanTo is set to a value of 0, 1, or 3.

See Also

ScanTo property

Methods (Scan control)

AboutBox

CloseScanner

OpenScanner

ResetScanner

ScannerAvailable

ShowScannerSetup

ShowSelectScanner

StartScan

ShowScanNew

ShowScanPage

StopScan

CloseScanner Method

Closes the scanner by unloading the driver.

Applies To

Image Scan control

Syntax

object.**CloseScanner**

Arguments

None

Return Value

Long

Most errors need to be intercepted by Catch ( ) in C+
+ or by On Error... in Visual Basic.

A busy condition returns an informational message.

Remarks

When a scanner or application error occurs, close the scanner before you access it again to avoid
possible problems.

See Also

OpenScanner method

OpenScanner Method

Opens the scanner by loading the driver.

Applies To

Image Scan control

Syntax

object.**OpenScanner**

Arguments

None

Return Value

Long

Most errors need to be intercepted by Catch ( ) in C+
+ or by On Error... in Visual Basic.

A busy condition returns an informational message.

See Also

CloseScanner method

ResetScanner Method

Resets the scanner hardware and software.

Applies To

Image Scan control

Syntax

object.**ResetScanner**

Arguments

None

Return Value

Long

Most errors need to be intercepted by Catch ( ) in C+ + or by On Error... in Visual Basic.

A busy condition returns an informational message.

ScannerAvailable Method

Checks to see if any TWAIN-compatible software is available.

Applies To

Image Scan control

Syntax

object.**ScannerAvailable**

Return Value

Boolean

Setting	Description
True	Support for a TWAIN-compliant scanner is available.
False	Support for a TWAIN-compliant scanner is not available.

Remarks

This is done by checking the software (drivers and .dlls), not the hardware.

See Also

ShowScannerSetup method

ShowScannerSetup Method

Displays the setup dialog box provided by the TWAIN scanner driver.

Applies To

Image Scan control

Syntax

object.**ShowScannerSetup**

Arguments

None

Return Value

Long

Most errors need to be intercepted by Catch ( ) in C+
+ or by On Error... in Visual Basic.

A busy condition returns an informational message.

Remarks

The dialog box and selectable options are provided by the library twain.dll.

ShowSelectScanner Method

Select the TWAIN scanner you want to use.

Applies To

Image Scan control

Syntax

object.**ShowSelectScanner**

Arguments

None

Return Value

Long

Most errors need to be intercepted by Catch ( ) in C+ + or by On Error... in Visual Basic.

A busy condition or cancel returns an informational message.

Remarks

Displays a dialog box that shows the installed TWAIN scanners.

See Also

ScannerAvailable method

StartScan Method

Starts the scanning process.

Applies To

Image Scan control

Syntax

object.**StartScan**

Arguments

None

Return Value

Long

Most errors need to be intercepted by Catch ( ) in C+
+ or by On Error... in Visual Basic.

A busy condition or cancel returns an informational
message.

Remarks

If the scanner is closed, it will be opened.

See Also

Image property

PageOption property

ScanTo property

MultiPage property

FileType property

ShowScanNew Method

Displays the Scan New dialog box from which you can change options and scan.

Applies To

Image Scan control

Syntax

object.**ShowScanNew**[={*True*|*False*}]

Arguments

Boolean (optional)

Variant

Setting	Description
True (default)	Modal dialog.
False	Modeless dialog.

Return Value

Long

Most errors need to be intercepted by Catch ( ) in C++ or by On Error... in Visual Basic.

A busy condition or cancel returns an informational message.

Remarks

Provides an interactive method to change scanning options before starting a session.

The scanner will be opened if it is not already.

ShowScanPage Method

Displays the Scan Page dialog box from which you can change options and scan.

Applies To

Image Scan control

Syntax

object.**ShowScanPage**[={*True*|*False*}]

Arguments

Boolean (optional)

Variant

Setting	Description
True (default)	Modal dialog.
False	Modeless dialog.

Return Value

Long

Most errors need to be intercepted by Catch ( ) in C+ + or by On Error... in Visual Basic.

A busy condition or cancel returns an informational message.

Remarks

This method is used to Insert, Append, or Overwrite pages in an existing image file while scanning.

StopScan Method

The scanning operation is halted.

Applies To

Image Scan control

Syntax

object.**StopScan**

Arguments

None

Return Value

Long

Most errors need to be intercepted by Catch ( ) in C+
+ or by On Error... in Visual Basic.

A busy condition returns an informational message.

Remarks

The scan operation will stop on the page that is currently being scanned. Cleanup, including clearing paper and buffers, will be performed.

Events (Scan control)

The Scanner control uses the following custom Events:

PageDone

ScanDone

ScanStarted

A standard Event, Error, is used for fired errors.

PageDone Event

Signals the completion of each scanned page.

Applies To

Image Scan control

Syntax

*object*_**PageDone**

Remarks

This event is triggered after a page is scanned.

Return Value

Long

Passes the Number of the page
scanned.

ScanDone Event

Signals the end of a scanning operation.

Applies To

Image Scan control

Syntax

*object*_**ScanDone**

Remarks

This event is triggered immediately after a scanning operation is completed.

ScanStarted Event

Signals the start of a scanning operation.

Applies To

  Image Scan control

Syntax

  *object*_**ScanStarted**

Remarks

  This event is triggered just before scanning starts.

Wang Image Thumbnail Control

The Image Thumbnail control displays thumbnail views of the images contained in an image file in uniformly sized and spaced frames called thumbnail boxes. Thumbnails are useful for displaying each page of a multipage image file in a miniature format. When you specify an image file, thumbnail representations of images are generated and buffered for use. You can use the buffered thumbnails or choose to regenerate thumbnails to reflect subsequent changes made to the image file, such as inserted or deleted pages, or alterations made to pages (annotations).

The Thumbnail control can be configured to scroll either vertically or horizontally. When scrolled vertically, thumbnails are displayed sequentially from the top left, across the rows and down to the bottom right. When scrolled horizontally, thumbnails are displayed sequentially from the top left, down the columns and across to the bottom right. The scroll bar is displayed only when there are more thumbnails available than can fit in the control's window. A scrollbar is never displayed in the non-scrolling direction.

You can change the size of the thumbnail boxes collectively, making them smaller, to fit more into the control's window, or making them larger to see more detail. The views displayed in the thumbnail boxes maintain the same aspect ratio (height to width relationship) as the image pages they represent.

Although the Thumbnail control can only display views of the image, it can be used in conjunction with the Admin control to add or delete pages in the image file.

You can also choose to display text labels beneath the thumbnails, and assign different fonts, colors, and styles to the labels.

See Also

Imaging Concepts

Properties (Thumbnail control)

BackColor

BorderStyle

Enabled

FirstSelectedThumb

HighlightColor

HighlightSelectedThumbs

hWnd

Image

LastSelectedThumb

MouseIcon

MousePointer

ScrollDirection

SelectedThumbCount

StatusCode

ThumbBackColor

ThumbCaption

ThumbCaptionColor

ThumbCaptionFont

ThumbCaptionStyle

ThumbCount

ThumbHeight

ThumbSelected[PageNumber]

ThumbWidth

BackColor Property

Returns or sets the background color of the entire control.

Applies To

Image Thumbnail control

Syntax

object.**BackColor**[=*color*]

Data Type

Long

Remarks

Use the RGB format. The default value is light gray -- RGB (192, 192, 192).

The color specified should be different from the color set by the **HighlightColor** property to ensure the highlight remains visible against the background.

Do not confuse this property with the **ThumbBackColor** property, which sets the color of the background of the thumbnail boxes.

See Also

HighlightColor property

ThumbBackColor property

BorderStyle Property

Returns or sets if the control is displayed with or without a border.

Applies To

Image Thumbnail control

Syntax

*object.***BorderStyle**[*=value*]

Data Type

Integer (enumerated)

Setting	Description
0	No border is displayed.
1 (default)	A single line, fixed size border is displayed.

Remarks

This property can only be set at design time.

FirstSelectedThumb Property

Returns the page number of the first selected thumbnail.

Applies To
  Image Thumbnail control

Syntax
  *object.***FirstSelectedThumb**

Data Type
  Long

Remarks
  Available at run-time as read-only.
  The value will be in the range 0 (zero) to ThumbCount, inclusive.
  A value of 0 (zero) indicates there are no thumbnails selected.

HighlightColor Property

Returns or sets the color of the highlight that is applied to selected thumbnail boxes.

Applies To

Image Thumbnail control

Syntax

*object.***HighlightColor**[*=color*]

Data Type

Long

Remarks

The highlight appears only if the **HighlightSelectedThumbs** property is set to True.

Use the RGB format. The default value is black -- RGB (0, 0, 0).

The color specified should be different from the color set by the **BackColor** property to ensure the highlight remains visible against the background.

See Also

BackColor property

ThumbBackColor property

HighlightSelectedThumbs property

HighlightSelectedThumbs Property

Returns or sets whether a highlight will be displayed around selected thumbnails.

Applies To

Image Thumbnail control

Syntax

object.**HighlightSelectedThumbs**[={*True*|*False*}]

Data Type

Boolean

Setting	Description
True (default)	Selected thumbnails highlight is displayed.
False	Selected thumbnails highlight is not displayed.

Image Property

Returns or sets which image file's thumbnails are displayed.

Applies To

Image Thumbnail control

Syntax

*object.***Image**[*=filename*]

Data Type

String

Remarks

An empty string is a valid value that resets the control so that no thumbnails are displayed.

Specifying a new image or respecifying the current image forces the following changes:

The thumbnails of the image are displayed with the thumbnail of the first page located in the upper-left corner of the control.

SelectedThumbCount property value is reset to (0).

FirstSelectedThumb property value is reset to (0).

LastSelectedThumb property value is reset to (0).

ThumbCount property value is reset to the number of pages in the specified file.

Each element in the **ThumbSelected[PageNumber]** array is reset to False.

See Also

Images
SelectedThumbCount property
ThumbCount property
FirstSelectedThumb property
LastSelectedThumb property
ThumbSelected[PageNumber] property

LastSelectedThumb Property

Returns the page number of the last selected thumbnail.

Applies To

Image Thumbnail control

Syntax

*object.*__LastSelectedThumb__

Data Type

Long

Remarks

Available at run-time as read-only.

The value will be in the range 0 (zero) to ThumbCount, inclusive.

A value of 0 indicates there are no thumbnails selected.

MouseIcon Property

Returns or sets the cursor displayed when a custom MousePointer is used.

Applies To

Image Thumbnail control

Syntax

*object.***MouseIcon**[=Picture]

Data Type

Picture

Remarks

Only ICO and CUR file types are supported.

See Also

MousePointer property

MousePointer Property

Returns or sets the type of mouse cursor displayed when the cursor is positioned over the control.

Applies To

  Image Thumbnail control

Syntax

  *object.***MousePointer**[=*integer*]

Data Type

  Integer (enumerated)

Setting	Description
0 (default)	Default (For this control, the default is 'Arrow')
1	Arrow
2	Cross
3	I-Beam
4	Arrow
5	Size
6	Size NESW
7	Size NS
8	Size NW SE
9	Size W E
10	Up Arrow
11	Hourglass
12	No Drop
13	Arrow and Hourglass
14	Arrow and Question mark
15	Size All
99	Custom pointer specified by this property

See Also

  MouseIcon property

SelectedThumbCount Property

Returns the number of thumbnails currently selected.

Applies To

Image Thumbnail control

Syntax

*object.***SelectedThumbCount**

Data Type

Long

Remarks

Available at run-time as read-only.

The value will be in the range 0 (zero) to **ThumbCount**, inclusive.

A value of 0 (zero) means that no thumbnails are currently selected. This is the default value set each time a new image is specified by the **Image** property.

See Also

Image property

ThumbBackColor Property

Returns or sets the background color of the thumbnail boxes.

Applies To

Image Thumbnail control

Syntax

*object.***ThumbBackColor**[=*color*]

Data Type

Long

Remarks

Use the RGB format. The default value is dark gray -- RGB (128, 128, 128).

Do not confuse this property with the **BackColor** property, which sets the color for the control's entire background.

See Also

BackColor property

HighlightColor property

ThumbCaption Property

Returns or sets the optional custom caption used with the ThumbCaptionStyle property.

Applies To

Image Thumbnail control

Syntax

*object.***ThumbCaption**[=*caption*]

Data Type

String

Remarks

Lets you customize the caption beneath each thumbnail.

Used when the **ThumbCaptionStyle** property specifies that captions are to be displayed.

You can use the **ThumbCaptionFont** property and **ThumbCaptionColor** property to alter the appearance of captions.

You can use the reserved characters # (number sign) to represent the thumbnail's page number, and * (asterisk) to represent the total number of thumbnails. For example, the captions 'Page 1 of 20', 'Page 2 of 20', and so on, can be displayed by specifying the string 'Page # of *'.

See Also

ThumbCaptionStyle property

ThumbCaptionFont property

ThumbCaptionColor property

ThumbCaptionColor Property

Returns or sets the color of caption labels displayed beneath thumbnails.

Applies To

Image Thumbnail control

Syntax

*object.***ThumbCaptionColor**[*=color*]

Data Type

Long

Remarks

Use the RGB format. The default value is black -- RGB (0, 0, 0).

You can use the **ThumbCaptionFont** property and **ThumbCaptionStyle** property to alter the appearance of captions.

See Also

ThumbCaptionFont property

ThumbCaptionStyle property

ThumbCaptionFont Property

Returns or sets the font used to display captions.

Applies To

Image Thumbnail control

Syntax

*object.***ThumbCaptionFont**[*=font*]

Data Type

OLEFONT

Remarks

The default font is MS Sans Serif, regular weight, 12 point; no italic, underline, or strikethrough.

See Also

ThumbCaptionColor property

ThumbCaptionStyle property

ThumbCaptionStyle Property

Returns or sets the type of caption that appears beneath each thumbnail.   You can also choose not to display captions.

Applies To

   Image Thumbnail control

Syntax

   *object.***ThumbCaptionStyle**[*=integer*]

Data Type

   Integer (enumerated)

Setting	Description
0	No captions beneath the thumbnails.
1 (default)	Page number, left justified, beneath each thumbnail.
2	Same as Setting 1, but also includes a graphic before the caption for those thumbnail pages whose image file pages have annotations present.
3	The string set by the property **ThumbCaption** is displayed, left justified, beneath the thumbnail.
4	Same as Setting 3, but also includes a graphic before the caption for those thumbnail pages whose image file pages have annotations present.

Remarks

   You can use the **ThumbCaptionColor** and **ThumbCaptionFont** properties to alter the appearance of captions.

   Note that any captions displayed are clipped to the width of the thumbnail box.

See Also

   ThumbCaption property
   ThumbCaptionColor property
   ThumbCaptionFont property

ThumbCount Property

Returns the total number of pages present in the current image file.

Applies To

Image Thumbnail control

Syntax

*object.***ThumbCount**

Data Type

Long

Remarks

Available at run-time as read-only.

ThumbCount is recalculated each time a valid image is specified, or when either of the methods **InsertThumbs** or **DeleteThumbs** is executed.

See Also

Image property

InsertThumbs method

DeleteThumbs method

ThumbHeight Property

Returns or sets the height of thumbnail boxes.

Applies To

Image Thumbnail control

Syntax

*object.***ThumbHeight**[=*value*]

Data Type

Long

Remarks

Used with the **ThumbWidth** property to specify the size of the displayed thumbnail boxes.

Specify a height between 50 and 500 (pixels).

The default value of 110, used with the ThumbWidth property default value of 85, produces a close representation of a standard letter-sized sheet.

If the thumbnail box size is altered, new thumbnail representations must be generated for all pages of the specified image file. This is equivalent to calling the **ClearThumbs** method or respecifying the **Image** property.

See Also

ThumbWidth property

UISetThumbSize property

ThumbSelected[PageNumber] Property

Returns or sets the selection status of the specified thumbnail.

Applies To

Image Thumbnail control

Syntax

*object.***ThumbSelected**[*PageNumber*][={*True*|*False*}]

Data Type

| ThumbSelected | Boolean |
| PageNumber | Long |

ThumbSelected

Setting	Description
True	The thumbnail is selected.
False (default for each item in the array)	The thumbnail is not selected.

Remarks

Valid PageNumber values must be between 1 and the current value of the **ThumbCount** property (inclusive).

If the **HighlightSelectedThumbs** property is set to True, selected thumbnails will display a border of the color set in the **HighlightColor** property.

See Also

HighlightSelectedThumbs
FirstSelectedThumb property
LastSelectedThumb property
SelectedThumbCount property
HighlightColor property
SelectAllThumbs method
DeselectAllThumbs method

ScrollDirection Property

Returns or sets the scrolling direction.

Applies To

Image Thumbnail control

Syntax

*object.***ScrollDirection**[= {0|1}]

Data Type

Integer (enumerated)

Setting	Description
0	Displays a horizontal scrollbar.
1 (default)	Displays a vertical scrollbar.

Remarks

The scrollbar will be displayed in the direction selected, when required, to accommodate more thumbnails.

Note that either horizontal or vertical scroll may be specified, but not both.

ThumbWidth Property

Returns or sets the width of the thumbnail boxes.

Applies To

Image Thumbnail control

Syntax

*object.***ThumbWidth**[=*value*]

Data Type

Long

Remarks

Used with the ThumbHeight property to specify the size of the displayed thumbnail boxes.

Specify a width between 50 and 500 (pixels).

The default value of 85, used with the **ThumbHeight** default value of 110, produces a close representation of a standard letter sized sheet.

If the thumbnail box size is altered, new thumbnail representations must be generated for all pages of the specified image file. This is equivalent to calling the **ClearThumbs** method or respecifying the **Image** property

See Also

ThumbHeight property

UISetThumbSize method

Methods (Thumbnail control)

AboutBox

ClearThumbs

DeleteThumbs

DeselectAllThumbs

DisplayThumbs

GenerateThumb

GetMaximumSize

GetMinimumSize

GetScrollDirectionSize

InsertThumbs

Refresh

ScrollThumbs

SelectAllThumbs

UISetThumbSize

ClearThumbs Method

Clears the buffered thumbnail representations of one or all pages.

Applies To

Image Thumbnail control

Syntax

object.**ClearThumbs** *pagenumber*

Arguments

Parameter	Data Type	Description
PageNumber	Long	Optional. If a value is not specified, a default value of 0 (zero) is assumed. The value 0 clears all pages.
		The one (1) relative page number that corresponds to the page whose thumbnail will be cleared.

Remarks

Thumbnails that have been cleared will be regenerated either when needed for display, or by calling the **GenerateThumb** method.

DeleteThumbs Method

Notifies the Thumbnail control that pages have been deleted from the image whose thumbnails are currently displayed.

Applies To

Image Thumbnail control

Syntax

object.**DeleteThumbs** *DeleteAt, DeleteCount*

Arguments

Parameter	Data Type	Description
DeleteAt	Long	The one (1) relative page number of the first page deleted. Valid values are in the range 1 to ThumbCount.
DeleteCount	Long	Optional. The number of pages to be deleted. If a value is not specified, a default value of 1 is used.

Remarks

The pages must be removed from the image file itself prior to calling this method. This can be done by first calling the Admin control's **DeletePages** method.

DeleteThumbs preserves the integrity of the control's internal mapping of displayed thumbnails and the pages of the specified image file.   If pages are deleted from the displayed image, and this method is not called, the thumbnails displayed will not reflect the change.

This method refreshes the control's display.

See Also

DeletePages method, Wang Image Admin control

DeselectAllThumbs Method

Deselects all thumbnails.

Applies To

Image Thumbnail control

Syntax

object.**DeselectAllThumbs**

Remarks

This method is equivalent to setting each item in the **ThumbSelected[ ]** array to False.

See Also

ThumbSelected[PageNumber] property

DisplayThumbs Method

Displays thumbnails using specified options.

Applies To

Image Thumbnail control

Syntax

object.**DisplayThumbs** *ThumbNumber, Option*

Arguments

Parameter	Data Type	Description
ThumbNumber	Long	Optional. The number of the Thumbnail whose position will be set. If a value is not specified, a default value of 1 is used, and the Option parameter cannot be specified.
Option	Integer	Optional. Sets the position of the specified Thumbnail within the control. If no value is specified, a default value of 0 (zero) is used. Allowable values are:

Setting	Description
0	Thumbnail box is displayed in the top row (for vertical scrolling).
0	Thumbnail box is displayed in leftmost column (for horizontal scrolling).
1	Thumbnail box is displayed in the middle row.
2	Thumbnail box is displayed in the bottom row (for vertical scrolling).
2	Thumbnail box is displayed in the rightmost column (for horizontal scrolling).

Remarks

This method refreshes the control's display.

Note that it is not necessary to call this method after setting the **Image** property to initiate a display of the specified image file's thumbnails.

GenerateThumb Method

Forces the generation or regeneration of the specified image's thumbnail representation.

Applies To

Image Thumbnail control

Syntax

object.**GenerateThumb** *Option, PageNumber*

Arguments

Parameter	Data Type	Description
Option	Integer	Indicates when the Thumbnail of a specified page should be generated. Allowable values are:

Setting	Description
0	Generates a thumbnail if one has not already been obtained.
	Not specifying a PageNumber, or specifying a PageNumber of 0 causes all pages that have not been obtained to be generated.
1	Causes the thumbnail to be generated immediately. If the thumbnail was previously generated, a new one will be obtained.
	Not specifying a PageNumber, or specifying a PageNumber of 0 causes all pages that have not been obtained to be generated.

PageNumber	Long	Optional. The number of the page for which a Thumbnail will be generated. If a value is not specified, a default value of 0, to indicate that ALL pages are to be generated, is used.

Remarks

This method should be called to reflect any changes made to the image file after a previous thumbnail was generated.

It is not always necessary to obtain new thumbnails (Option=1). Thumbnail images that have been generated during the current session remain buffered, even though they may not be displayed.

Use caution when using a PageNumber of 0. Resulting operations can be lengthy.

GetMaximumSize Method

Determines the largest non-scrolling control dimension that can be used to display a specified number of thumbnail boxes.

Applies To

Image Thumbnail control

Syntax

*object.***GetMaximumSize**(*ThumbCount, ScrollBar*)

Arguments

Parameter	Data Type	Description
ThumbCount	Long	The number of thumbnail boxes desired in the non-scrolling direction.
ScrollBar	Boolean	Indicates whether or not the size value returned allows space for a scroll bar.
		True -- Allows space for a scroll bar. False -- Does not allow space for a scroll bar.

Return Value

Long

The maximum control size, in pixels, needed to accommodate the number of thumbnail boxes specified by the ThumbCount parameter.

Remarks

GetMaximumSize uses the current values of the **ScrollDirection**, **ThumbHeight**, and **ThumbWidth** properties to calculate this value.

GetMaximumSize and **GetMinimumSize** should be used together to determine a range of control sizes that allow for a display that fits the required number of thumbnails in the non-scrolling dimension.

A control size larger than this dimension in the non-scrolling direction will contain space for more than the specified number of thumbnails.

Once a control size is determined (based on the results of GetMaximumSize and GetMinimumSize), a call to the **GetScrollDirectionSize** method may be used to calculate the size required in the scrolling direction to display a specified number of thumbnails.

See Also

GetMinimumSize method

GetScrollDirectionSize method

ScrollDirection property

ThumbHeight property

ThumbWidth property

GetMinimumSize Method

Determines the smallest non-scrolling control dimension that can be used to display a specified number of thumbnail boxes.

Applies To

Image Thumbnail control

Syntax

object.**GetMinimumSize**(*ThumbCount, ScrollBar*)

Arguments

Parameter	Data Type	Description
ThumbCount	Long	The number of thumbnail boxes desired in the non-scrolling direction.
ScrollBar	Boolean	Indicates whether or not the size value returned allows space for a scroll bar.
		True -- Allows space for a scroll bar. False -- Does not allow space for a scroll bar.

Return Value

Long

The minimum control size needed to accommodate the number of thumbnail boxes specified by the ThumbCount parameter.

Remarks

GetMinimumSize uses the current values of the **ScrollDirection**, **ThumbHeight**, and **ThumbWidth** properties to calculate this value.

GetMaximumSize and GetMinimumSize should be used together to determine a range of control sizes that allow for a display that fits the required number of thumbnails in the non-scrolling dimension.

A control size smaller than this dimension in the non-scrolling direction will contain space for fewer than the specified number of thumbnails.

Once a control size is determined (based on results of GetMaximumSize and GetMinimumSize), a call to the **GetScrollDirectionSize** method may be used to calculate the size required in the scrolling direction to display a specified number of thumbnails.

See Also

GetMaximumSize method

GetScrollDirectionSize method

ScrollDirection property

ThumbHeight property

ThumbWidth property

GetScrollDirectionSize Method

Determines what the control size has to be, in the scrolling dimension, to display a specified number of thumbnails.

Applies To

Image Thumbnail control

Syntax

object.**GetScrollDirectionSize**(*ScrollDirectionThumbCount, NonScrollDirectionThumbCount, NonScrollDirectionSize, ScrollBar*)

Arguments

Parameter	Data Type	Description
ScrollDirectionThumbCount	Long	The number of thumbnail boxes desired in the scrolling direction.
NonScrollDirectionThumbCount	Long	The number of thumbnail boxes desired in the non-scrolling direction.
NonScrollDirectionSize	Long	The size of the control's area, in pixels, in the non-scrolling dimension for which a corresponding scroll direction size is desired.
ScrollBar	Boolean	Indicates whether or not the size specified for NonScrollDirectionSize allows space for a scroll bar. True -- Allows space for a scroll bar. False -- Does not allow space for a scroll bar.

Return Value

Long

> Returns the control size, in pixels, needed to accommodate the number of thumbnail boxes specified by the parameter ScrollDirectionThumbCount in the scrolling direction.

Remarks

If NonScrollDirectionSize is determined by first calling the **GetMaximumSize** and **GetMinimumSize** methods, **GetScrollDirectionSize** can calculate the size and scrolling direction required to display a specified number of thumbnails.

The value of the argument NonScrollDirectionThumbCount must be the same value passed as the argument ThumbCount to the GetMaximumSize and GetMinimumSize methods.

The value of the argument ScrollBar must be the same in the GetMaximumSize, GetMinimumSize, and GetScrollDirectionSize methods.

See Also

GetMinimumSize method

GetMaximumSize method

ThumbWidth property

ThumbHeight property

ScrollDirection property

InsertThumbs Method

Informs the Thumbnail control that pages have been inserted into the currently displayed image file.

Applies To

Image Thumbnail control

Syntax

object.**InsertThumbs** *InsertBeforeThumb, InsertCount*

Arguments

Parameter	Data Type	Description
InsertBeforeThumb	Long	Optional. The page number of the thumbnail before which the new thumbnails will be inserted.
		A value of 1 means the thumbnails will be inserted before the first page.
		A value of **ThumbCount**+1 means the thumbnails will be inserted after the last page.
		If InsertBeforeThumb is not specified, the default value of **ThumbCount**+1 is used, and the InsertCount parameter cannot be set.
InsertCount	Long	Optional. The number of pages inserted.
		If a value is not specified, the default value of 1 is used.

Remarks

The pages must be inserted into the image file itself prior to calling this method. This can be done by first calling the Admin control's **Insert** method.

InsertThumbs preserves the integrity of the control's internal mapping of displayed thumbnails and the pages of the specified image file. If pages are inserted into the displayed image file and this method is not called, the thumbnails displayed will not reflect the change.

This method refreshes the control's display.

See Also

ThumbCount property

InsertThumbs method

Insert method, Wang Image Admin control

Refresh Method

Repaints the control.

Applies To

Image Thumbnail control

Syntax

object.**Refresh**

Arguments

None

Remarks

If the control has an active window, the refresh occurs synchronously. Otherwise, the control is refreshed asynchronously.

This method refreshes the control's display.

ScrollThumbs Method

Specifies a direction and amount for scrolling.

Applies To

Image Thumbnail control

Syntax

object.**ScrollThumbs**(*Direction, Amount*)

Arguments

Parameter	Data Type	Description
Direction	Integer	Specifies the direction of scroll.
		A value of 0 (zero) scrolls the control towards the last thumbnail.
		A value of 1 scrolls the control towards the first thumbnail.
Amount	Integer	Specifies the amount of scroll desired.
		A value of 0 (zero) scrolls in the requested direction by almost a full control screenful. A small amount of original screen area is left to serve as a reference.
		The amount scrolled is equivalent to the amount scrolled when you click in the scroll bar (between the up / down arrows and the scroll box).
		A value of 1 scrolls the control in the requested direction by a small amount.
		The amount scrolled is equivalent to the amount scrolled when you click the scroll bar's up or down arrow box once.

Return Value

Boolean

The return value indicates if the control was scrolled.

Setting	Description
True	The scroll was performed.
False	The scroll was not performed (for example, if the scroll was at either end of its scrolling limit and could not scroll any further).

See Also

ScrollDirection property

SelectAllThumbs Method

Selects all thumbnails.

Applies To

Image Thumbnail control

Syntax

object.**SelectAllThumbs**

Remarks

This method is equivalent to setting each item in the **ThumbSelected[ ]** array to True.

See Also

ThumbSelected[ ] property

UISetThumbSize Method

Displays a dialog box from which a user can set the width and height of the thumbnail box.

Applies To

Image Thumbnail control

Syntax

object.**UISetThumbSize**(*Image, PageNumber*)

Arguments

Parameter	Data Type	Description
Image	String	Optional. Used with the PageNumber parameter to specify the name and page of the image file to be displayed in the sample thumbnail.
		If not specified, the default value of an empty string will be used, and an image will not be displayed.
PageNumber	Long	Optional. Used with the Image parameter to specify the name and page of the image file to be displayed in the sample thumbnail.
		If a value is not specified, a default value of 1 is used.

Return Value

Boolean

Setting	Description
True	The dialog box was dismissed with the OK button, generally indicating new values have been set for the **ThumbHeight** and **ThumbWidth** properties.
False	The dialog box was dismissed with the CANCEL button, indicating the values for the ThumbHeight and ThumbWidth properties are unchanged.

Remarks

If the thumbnail box size is altered, new thumbnail representations must be generated for all pages of the specified image file. This is equivalent to calling the **ClearThumbs** method or respecifying the **Image** property.

See Also

ClearThumbs method

Events (Thumbnail control)

The Events used by the Thumbnail control are all standard.

*Click	*MouseDown
*DblClick	*MouseMove
KeyDown	*MouseUp
KeyUp	

Events with an asterisk (*) use the additional argument ThumbNumber.

Argument

Parameter	Data Type	Description
ThumbNumber	Long	The number of the thumbnail on which the event occurred. This parameter is 1 relative (that is, a value of 1 means the event occurred on the thumbnail for page 1).
		A 0 (zero) value means an event occurred on the control, but not on a thumbnail.