ImageMan/VB Image Control

Properties Events

Description

The image control provides the ability to load, display, print, process and save images in JPEG, TIFF, BMP, DIB, PCX, WMF, Targa, DCX, GIF, IMG and EPS formats from your Visual Basic application.

Custom Image properties

About
AppendImage
AutoDraw
AutoScale
Brightness
Blue
ClipboardCommand
Compression
DataChanged
DataField
DataSource
DisplayColors
DitherMethod
DoPrint
DstLeft
DstTop
DstRight
DstBottom
Ext
Extensions
ExtensionCount
Gamma
GetFilename
Green
hDC
hDIB
hImage
hImageData
hPalette
ImageDataSize
IncreaseTo
Invert
ImageColors
ImageFlags
ImageHeight
ImageWidth
ImageXRes
ImageYRes
Magnification
MemoryFormat
Mirror
Overwrite
Pages
PageNumber
PaletteEntries
Picture
PrnHDC
Quality
Red
ReduceTo
Rotate
SaveAs
SaveOptions
ScaleLeft
ScaleMethod
ScaleTop
ScaleHeight
ScaleWidth
Scrollbars
SrcLeft
SrcTop
SrcRight
SrcBottom
UseDDB
VBPicture

Custom Image Events

Paint Scroll Select Status

ImageMan/VBX Technical Support

You may obtain technical support for the ImageMan/VBX controls by Email, Phone, FAX, CompuServe and our own StarMan Bulletin Board system.

Data Techniques, Inc.
300 Pensacola Road
Burnsville, NC 28714

Internet: support@data-tech.com
Web: www.data-tech.com
FTP: ftp.data-tech.com
Tech Support: 704-682-4111 9am to 5pm EST
Fax: 704-682-0025
BBS: 704-682-4356 (28.8 V.34)
Compuserve: GO DATATECH

To get automatic notifications of updates and other important product information, join the ImageMan Mailing List. Send email to: *ImageMan-Request@data-tech.com* with the word *subscribe* in the message body.

About Property

Description

At development time displays a dialog containing the version and copyright of the control.

AppendImage Property

Description
When saving an image to a TIFF or DCX file this controls whether the image is appended to an existing file or the file is overwritten.

Usage
ImageControl.**AppendImage**[= setting%]

Remarks
The AppendImage property settings are:

Setting	Description
True	Append image when saving to TIFF or DCX file formats
False	Do not append. Will overwrite file based on value of Overwrite property.

Data Type
Boolean

AutoScale Property

Description
Determines whether the control will be redrawn automatically when a new image is loaded.

Usage
ImageControl.**AutoScale**[= setting%]

Remarks
The AutoScale property settings are:

Setting	Description
0	Do not scale the image to fit into the control.
1	Scale the image to fit into the control and maintain the aspect ratio.
2	Stretch the image to fit into the control.

Note
Setting this property to a value of 1 or 2 will override the Magnification and ScaleWidth/ScaleHeight properties.

Data Type
Integer(Enumerated)

AutoDraw Property

Example

Description
Determines whether the control will be redrawn automatically when a new image is loaded.

Usage
ImageControl.**AutoDraw**[= Bool%]

Remarks
The AutoDraw property settings are:

Setting	Description
True(-1)	The control will be redrawn when a new image is loaded.
False(0)	The control must be redrawn by calling the Refresh method after loading an image.

Note
Setting this property to **False** allows you to adjust the Magnification or Scale properties before the image is displayed. When set to **True** the image will be drawn and scaled to fit the control.

Data Type
Integer(Boolean)

AutoDraw Property Example


```
' Load an image and use default scaling
ImageMan1.AutoDraw = True
ImageMan1.Picture = "c:\sample.tif"

' Load an image and set the Magnification to 50%
ImageMan1.AutoDraw = False
ImageMan1.Picture = "c:\sample.tif"    ' Image Loaded but not displayed
ImageMan1.Magnification = 50
ImageMan1.Refresh                      ' Draw the Image
```

Blue Property

Example

Description
Sets the blue value for a color entry in the image's palette.

Usage
ImageControl.**Blue**(Index%)[= Setting%]

Remarks
Use this property along with the Red and Green properties to change the colors in a 1, 16 or 256 color image. After changing these properties you must call the Refresh method to redraw the image with the new colors. The Index% value must be between 0 and the value of the PaletteEntries property. The allowable values for the property are 0 to 255.

Data Type
Integer Array

Red, Green, Blue Properties Example


```
' Darken an image by subtracting a value from each palette entry

Dim Idx%

If ImageMan1.ImageColors <= 256 Then        ' There's No Palette on 24
bit Images
     For Idx = 0 to ImageMan1.PaletteEntries - 1
          ImageMan1.Red(Idx) = ImageMan1.Red(Idx) - 10
          ImageMan1.Green(Idx) = ImageMan1.Green(Idx) - 10
          ImageMan1.Blue(Idx) = ImageMan1.Blue(Idx) - 10
     Next Idx
End If
```

Brightness Property

Description
Adjusts the brightness for an image.

Usage
ImageControl.**Brightness**[= Setting%]

Remarks
Setting Brightness to a value of 0 will display the image with the default brightness. Set this property to a number between 0 and -255 to lighten the image or set it to a number between 1 and 255 to darken it.

Data Type
Integer

Click Event

Description
Occurs when the user presses and then releases a mouse button over a control.

Syntax
Sub ctlname_Click(Index as Integer)

Remarks
The argument Index uniquely identifies a control in a control array.

Note
The Click procedure is only generated for left mouse button activity. Use the MouseDown and MouseUp events to handle other mouse buttons.

ClipboardCommand Property

Example

Description
Specifies the image should be copied from or to the Windows clipboard.

Usage
ImageControl.**ClipboardCommand** = 1 | 2

Remarks
Setting this property to a value of one (1) will cause the image or the specified portion to be copied to the clipboard. To specify a portion of an image set the SrcLeft, SrcTop, SrcRight and SrcBottom properties to the portion to be copied.

Setting this property to a value of two (2) will paste an image on the clipboard into the control. The clipboard must contain an image in CF_DIB or CF_METAFILE format for the operation to be successful.

Data Type
Integer (Enumerated)

Compression Property

Description
Determines what compression method will be used when saving an image using the SaveAs property.

Usage
ImageControl.**Compression**[= setting%]

Remarks
The Compression property settings are:

Setting	Description
0	No Compression.
1	LZW - Used in TIFF and GIF formats
2	Huffman - Used in TIFF
3	Packbits - Used in TIFF
4	Fax Group 3 - Used in TIFF
5	Fax Group 4 - Used in TIFF

Note

Some formats like GIFand PCX always store the images in compressed format so this property will be ignored.

If a compression method is selected using this property and the requested image format does not support that method then the format's default compression method will be used.

Data Type
Integer(Enumerated)

DataChanged Property

Description
Indicates that the image in the control has been changed by some process other than getting data from the current record. Not available at design time; read/write at runtime.

Usage
ImageControl.**DataChanged** [= {True | False}]

Remarks
The DataChanged property settings are:

Setting	Description
True	The image in the control is not the same as in the current record.
False	The image in the control, if any, is the same as the data in the current record.

Remark
When a data control moves from record to record, each bound control automatically displays the image associated with the current record and DataChanged is set to **False.** If the data in the control changes in any way other than moving to a different record, DataChanged becomes **True.**

When the data control starts to move to a different record, the Validate event occurs. Then if DataChanged is **True** for any bound control, the data control invokes the **Edit** and **UpdateRecord** methods. Finally, data from any control where the DataChanged property is set to **True** is saved to the database.
In the code for the validate event, you can set this property to **False** for any bound control where you do not want to save the image in the database.

Inspect the value of the DataChanged property in your code for a control's Change event to avoid a cascading event.

Data Type
Integer(Boolean)

DataField Property

Description
Binds an Image control to a field. Read/write at both design time and runtime.

Usage
ImageControl.**DataField**[= Fieldname$]

Remarks
Bound controls provide access to images store in your database. The DataSource property of a bound control specifies a valid data control name and the DataField property specifies the name of a text or binary field in your database. Together, these properties specify which data appears in the control.

If the field named by the DataField property is a text field then the ImageMan/VB control will store the filename of the current image in the field. If the field is a long binary field, then the control will store the image data in the field in compressed format.

Because ImageMan/VB uses compression when saving images into a database, those images can only be read by the ImageMan/VB control.

Data Type
String

DataSource Property

Description
Determines the data control through which the current control is bound to a database. Read/write at design time; not available at runtime.

Usage
ImageControl.**DataSource**[= DataCtrlName$]

Remarks
To bind a control to a field in a database at runtime, you must specify a data control in this property at design time. To complete the connection with a database, you must also provide the name of a Field object in the DataField property. Unlike the DataSource property, the DataField property can be provided at runtime.

Data Type
String

DblClick Event

Description
Occurs when the user presses and releases a mouse button, then presses it again over a control.

Syntax
Sub ctlname_DblClick(Index as Integer)

Remarks
The argument Index uniquely identifies a control in a control array.

Note
The DblClick procedure is only generated for left mouse button activity. Use the MouseDown and MouseUp events to handle other mouse buttons.

DisplayColors Property

Example

Description
Returns the number of colors supported by the current video driver mode.

Usage
ImageControl.**DisplayColors**

Note
This property can be used to determine if an image should be color reduced using the ReduceTo property.

Data Type
Single (Read-Only)

DisplayColors Example Code

Close Copy Print

```
' Load Image into ImageMan1 and then check to see if we need to color
reduce it
ImageMan1.Picture = "24bit.tif"

' Set the Dither Method to Floyd Steinberg in case we have to color
reduce it
ImageMan1.DitherMethod = 3

' Now check to see if the image has more colors than can be displayed
by
' the current video driver mode
if ImageMan1.ImageColors > ImageMan1.DisplayColors Then
     if ImageMan1.DisplayColors = 256 Then
          ImageMan1.ReduceTo = 4          ' 256 Color Optimized
Palette
     elseif ImageMan1.DisplayColors = 16 Then
          ImageMan1.ReduceTo = 5          ' 16 Color Fixed Palette
     endif
Endif

' Now draw the image
ImageMan1.Refresh
```

DitherMethod Property

Description
Sets or Returns the dither method which is used when reducing the number of colors in the image,

Usage
ImageControl.**DitherMethod** = 0 | 1 | 2 | 3

Remarks
The DitherMethod property settings are:

Setting	Description
NONE(0)	No Dithering - map to nearest color.
BAYER (1)	Selects the Bayer dither.
BURKES(2)	Selects the Burkes dither.
STEINBERG(3)	Selects the Floyd/Steinberg dither.

Note
This property selects which dither method is used when using the ReduceTo property to change the color format of an image.

Data Type
Integer(Enumerated)

DstLeft, DstTop Properties

Example

Description
Sets the coordinates of the upper left hand corner of the bounding rectangle used when printing an image.

Usage
ImageControl.**DstLeft** = left!
ImageControl.**DstTop** = top!

Remarks
The bounding rectangle defines the area on the page where the image will appear. These units are expressed in terms of the current printer scalemode.

Data Type
Single

Note
These values must be set before setting the DoPrint property to print the image.

DoPrint Property

Example

Description
Causes the image to be printed.

Usage
ImageControl.**DoPrint** = 1

Remarks
Setting this property to 1 causes the image to be printed in the bounding box specified by DstLeft, DstTop, DstRight, and DstBottom. . Before setting this property you must set the PrnHdc, DstLeft, DstTop, DstRight and DstBottom properties. To print only a portion of the image you must also set the SrcLeft, SrcTop, SrcRight, and SrcBottom properties.

Data Type
Integer(Boolean)

Note
The image will be printed at the resolution of the printer not that of the screen.

Printing An Image


```
' Print an Image at actual size centered on the page
' Declare the SetMapMode Windows API function
Declare Sub SetMapMode Lib "GDI" (ByVal hDC%, ByVal Mode% )
Global Const MM_TWIPS = 6

' The actual print code
Dim nPageXMid%, nPageYMid%
Dim ImgWidth%, ImgHeight%
Dim xRes%, yRes%

' Calculate the resolution of the image
' If the resolution isnt set then default to 300 DPI

' Calculate the horizontal res
if Imageman1.ImageXRes Then
      xRes = ImageMan1.ImageXRes
else
      xRes = 300
Endif

' Now calculate the vertical resoltion
if Imageman1.ImageYRes Then
      yRes = ImageMan1.ImageYRes
else
      yRes = 300
Endif


' Calculate the center of the page
nPageXMid = Printer.ScaleWidth / 2
nPageYMid = (Printer.ScaleHeight / 2) * -1


' Calculate the Image size in TWIPS
ImgWidth = ImageMan1.ImageWidth * 1440 / xRes
ImgHeight = ImageMan1.ImageHeight * 1440 / yRes

' Tell the control where to print the image
' We will center it on the page

ImageMan1.DstLeft = nPageXMid - ImgWidth / 2
ImageMan1.DstTop = nPageYMid + ImgHeight / 2
ImageMan1.DstRight = nPageXMid + ImgWidth / 2
ImageMan1.DstBottom = nPageYMid - ImgHeight / 2

' Initialize the VB printing mechanism by printing this empty string
Printer.Print ""
```

```
ImageMan1.PrnHdc = Printer.hDC

' Set the printer to the TWIPS logical mapping mode
' This ensure the image will be printed at the same size regardless of
the
' printer's resolution
SetMapMode Printer.hDC, MM_TWIPS

' Set the Hourglass Cursor, this may take a few seconds
MousePointer = 11
```

ImageMan1.DoPrint = 1

Printer.EndDoc

```
' Restore the Cursor
MousePointer = 0
```

DstRight, DstBottom Property

Example

Description
Sets the coordinates of the bottom right hand corner of the bounding rectangle used when printing an image.

Usage
ImageControl.**DstRight** = right!
ImageControl.**DstBottom** = bottom!

Remarks
The bounding rectangle defines the area on the page where the image will appear. These units are expressed in terms of the current printer scalemode.

Data Type
Single

Note
These values must be set before setting the DoPrint property to print the image.

ErrCode Property

Description
Returns a code containing the status of the last image operation.

Usage
ImageControl.**ErrCode**

Remarks
This property is set to zero (0) if the last operation completed successfully otherwise it contains the number of the error that occurred.

Data Type
Integer (Read-Only)

ErrString Property

Description
Returns a string containing the status of the last image operation.

Usage
ImageControl.**ErrString**

Remarks
If the ErrCode property is not zero then this property will return a string containing a textual description of the last error that occurred.

Data Type
String (Read-Only)

Ext Property

Description
Specifies the three letter extension of the image format which ImageMan/VB should use when opening images with non-standard extensions.

Usage
ImageControl.**Ext**[= ext$]

Remarks
Use this property only if you need to open images which have non-standard extensions. Set this property to an empty string to force ImageMan/VB to use the image's default extension. For instance to open a TIFF image which is called SAMPLE.001, you would set the Ext property to "TIF" and the Picture property to "SAMPLE.001."

Data Type
String (Read-Only)

ExtensionCount Property

Example

Description
Returns the number of image extensions supported by the control. This property is read-only.

Usage
ImageControl.**ExtensionCount**

Remarks
Use this property with the Extensions property to see a list of the supported image extensions.

Data Type
Integer (Read-Only)

Extensions Property

Example

Description
Returns all support image extensions for a control. This property is read-only.

Usage
ImageControl.**Extensions**(index%)

Remarks
This property works in conjunction with the ExtensionCount property which returns the number of supported extensions for the object.

This list is built when the ImageMan/VB is loaded. Because of ImageMan's object-oriented architecture the number of supported formats will vary based on which DIL files are installed.

Data Type
String Array (Read-Only)

ExtensionCount, Extension Properties Example

Close Copy Print

```
' Add the supported extensions to a listbox
Dim I%

For I = 0 to ImageMan1.ExtensionCount - 1
     List1.AdddItem   ImageMan1.Extensions(I)
Next I
```

Gamma Property

Description
When set to a value causes the image to be corrected by the specified gamma value.This property is write-only.

Usage
ImageControl.**Gamma** = GammaVal

Remarks
The allowable range for Gamma values is 1.0 to 5.0. After setting this property be sure to do a Refresh on the control to display the alterted image.

This property is only available at runtime.

Data Type
Single

GetFileName Property

Description
When set to a value of 1 causes the control's file open dialog to be displayed. This property is write-only.

Usage
ImageControl.**GetFileName** = 1

Remarks
This property should be set to 1 to cause the control to display the file open dialog. The dialog is the Windows 3.1 common file open dialog and is configured to list all image formats which the control can display. This property is only available at runtime.

Data Type
Integer(Boolean)

Green Property

Example

Description

Sets the green value for a color entry in the image's palette.

Usage

ImageControl.**Green**(Index%)[= Setting%]

Remarks

Use this property along with the Red and Blue properties to change the colors in a 256 color image. After changing these properties you must call the Refresh method to redraw the image with the new colors. The Index% value must be between 0 and the value of the PaletteEntries property. The allowable values for the property are 0 to 255.

Data Type

Integer Array

hDC Property

Example

Description
Returns a Windows Device Context for drawing into the image in the control.

Usage
ImageControl.**hDC**

Remarks

This property must be obtained for each image and allows you to use the Windows API functions to draw into the image. The changes made by using the hDC property change the internal bitmap maintained by the control. The altered image can be printed, saved or copied to the clipboard.

This property is read-only and available only at runtime.

Data Type
Integer

Using the hDC Property


```
Dim ImghDC

Declare Sub MoveTo Lib "GDI" ( ByVal hDC%, ByVal X%, ByVal Y% )
Declare Sub LineTo Lib "GDI" ( ByVal hDC%, ByVal X%, ByVal Y% )

' Draw an X through the image

ImghDC = ImageMan1.hDC

' These calls will draw a X over the image

MoveTo ImghDC, 0, 0
LineTo ImghDC, ImageMan1.ImageWidth, ImageMan1.ImageHeight

MoveTo ImghDC, ImageMan1.ImageWidth, 0
LineTo ImghDC, 0, ImageMan1.ImageHeight

' Update the image on screen

ImageMan1.Refresh

' Save the updated image
ImageMan1.SaveAs = "c:\altered.bmp"
```

hDIB Property

Description
Returns or Sets the handle of a Windows global memory block containing the image in DIB format.

Usage
ImageControl.**hDIB**[= hDib%]

Remarks

The application must free the handle when it is done by calling the Windows API function, GlobalFree, with the value of the memory block returned from the hDIB property. Each time this property is queried the control will allocate a new memory block containing the image in DIB format.

Data Type
Integer

hImage Property

Example

Description
Returns or Sets the internal ImageMan/VB handle for an image.

Usage
ImageControl.**hImage** [= hImage%]

Remarks
This property is used to copy images between ImageMan/VB controls. By using the DstRight and DstBottom properties the image can be scaled as it is copied into the new control. By default the image is copied at the same size.

The portion of the image that is copied can also be specified by using the SrcLeft, SrcTop, SrcRight and SrcBottom properties. By default, these properties are set to the entire image.

Note

Data Type
Integer

Using the hImage Property


```
' Make a 100x100 Thumbnail of the Image in ImageMan1 in ImageMan2
ImageMan1.DstRight = 100
ImageMan1.DstBottom = 100

ImageMan2.hImage = ImageMan1.hImage


' Resize the image in the control to half its original size
ImageMan1.DstRight = ImageMan1.ImageWidth / 2
ImageMan1.DstBottom = ImageMan1.ImageHeight / 2

ImageMan1.hImage = ImageMan1.hImage
```

ImageColors Property

Description
Returns the number of colors in the image. This property is read-only.

Usage
ImageControl.**ImageColors**

Data Type
Single

Note
This is the number of colors in the image not the number of colors the display driver supports.

hImageData Property

Example

Description
Specifies the Handle to a globally allocated block of memory from which image data will be read or has been written.

Usage
ImageControl.**hImageData** [= hMemoryBlock%]

Note
Using this property along with the ImageDataSize and MemoryFormat properties you can read and write compressed image data from memory.

When reading data from a memory block you must set the ImageDataSize property to the size of the image data in the allocated block whose handle you will set to the hImageData property.

When writing data to a block the Handle to the allocated memory block will be returned by the hImageData property and the size of the data within the block will be returned in the ImageDataSize property.

Data Type
Integer

hImageData Example Code

Close Copy Print

```
' This code assumes we have a handle to a global memory block which
contains an image in a
' format supported by ImageMan. We must also have the size of the
image data in the block.
' In most cases this is not the same as the size of the allocated
block since Windows will
' round up the size of the block to a specific boundary.

ImageMan1.ImageDataSize = 12000
ImageMan1.hImageData = hMyMemoryBlock

If ImageMan1.ErrCode <> 0 Then
      ' An error occurred, display a msg box with the error string
     MsgBox ImageMan1.ErrString, MB_OK, "ImageMan/VBX Error"
Endif

ImageMan1.Refresh

' The ImageMan control now contains the Image whose data was stored in
the passed in handle.

' It is the responsibility of the application to free the handle of
globally allocated memory
' passed to the hImageData property. The ImageMan control makes a copy
of the image data and ' ' does not free the handle which is passed in.


' Writing an image in a control to a memory block

' Set he MemoryFormat property to the extension of the format we wish
to save in
ImageMan1.MemoryFormat = "JPG"

' Now get the handle to the memory block
hMemBlock = ImageMan1.hImageData

' Now get the size of the image data
dwDataSize = ImageMan1.ImageDataSize

' The variable hMemBlock contains a handle to a globally allocated
memory block which contains
' the image data in the format specified by the MemoryFormat property.
You can also use the
' Compression, Quality, and SaveOptions properties to affect the saved
image.
' The application is responsible for freeing the allocated memory
```

```
block using the Windows API
' function, GlobalFree().
' The dwDataSize variable contains the size of the image data in the
memory block.
```

ImageDataSize Property

Example

Description
Specifies the size of a globally allocated block of memory from which image data will be read or has been written.

Usage
ImageControl.**ImageDataSize** [= MemoryBlkSize]

Note
Using this property along with the hImageData and MemoryFormat properties you can read and write compressed image data from memory.

When reading data from a memory block you must set this property to the size of the image data in the allocated block whose handle you will set to the hImageData property.

When writing data to a block the Handle to the allocated memory block will be returned by the hImageData property and the size of the data within the block will be returned in the ImageDataSize property.

Data Type
Integer

ImageFlags Property

Description
Returns a set of flags which describe the current image. This property is read-only.

Usage
ImageControl.**ImageFlags**

Remarks
Currently only the IMG_VECTOR (1) flag is supported. If this bit is set then the image is a vector image and certain operations cannot be performed on it. Currently the following operations cannot be performed on vector images:

- Color Reduction
- Rotation
- Palette Access
- Gamma & Brightness adjustment
- Getting an hDIB for the image

Data Type
Integer

Note

ImageHeight Property

Description
Returns the height of the image in image units. This is a read-only property.

Usage
ImageControl.**ImageHeight**

Data Type
Single

ImageWidth Property

Description
Returns the width of the image in image units. This is a read-only property.

Usage
ImageControl.**ImageWidth**

Data Type
Single

ImageXRes Property

Description
Returns the horizontal resolution of the image in dots per inch. This is a read-only property.

Usage
ImageControl.**ImageXRes**

Remarks
Some images may not contain resolution information therefore this property may be set to zero. Make sure to check for this condition before using this value.

Data Type
Single

ImageYRes Property

Description
Returns the vertical resolution of the image in dots per inch. This is a read-only property.

Usage
ImageControl.**ImageYRes**

Remarks
Some images may not contain resolution information therefore this property may be set to zero. Make sure to check for this condition before using this value. This value should be the same as the ImageXRes property in almost all images.

Data Type
Single

IncreaseTo Property

Description
Causes the color depth of an image to be increased to the specified bit depth. This property is write-only.

Usage
ImageControl.**IncreaseTo** = setting%

Remarks
The IncreaseTo property settings are:

Setting	Description
1	16 Colors (4 Bit)
2	256 Colors (8 Bit)
3	16 Million Colors (24 Bit)

Note
To decrease the number of colors in an image use the ReduceTo property.

Data Type
Integer(Enumerated)

Invert Property

Description
Setting this property causes the colors in the image to be inverted. This property is write-only.

Usage
ImageControl.**Invert** = 1

Remarks

Data Type
Integer (Write Only)

Magnification Property

Example

Description
Sets or Returns the percentage the image should be scaled by when being displayed.

Usage
ImageControl.**Magnification**[= Percent%]

Remarks
The property should be set to the percentage scaling desired, i.e. to scale the image by 50% set this property to 50. This property should be used when the image is to be scaled by the same percentage on both axes. If each axis needs a different scale percentage then the ScaleWidth and ScaleHeight properties should be used to scale the image.

When the ScaleWidth or ScaleHeight properties have been set manually then this property's value will be invalid.

This property does not scale the image only its screen representation. Use the hImage and Dst properties to scale the actual Image.

Data Type
Integer

MemoryFormat Property

Example

Description

Specifies the image format to be used when writing data to a memory block using the hImageData property and the format used when the control
is databound and linked to a binary field.

Usage

ImageControl.**MemoryFormat** [= ImageFormat$]

Note

Using this property along with the ImageDataSize and MemoryFormat properties you can read and write compressed image data from memory.

This property is also used to specify the format to be used when writing image data to a bound field in a database.

This property must be set to a valid image format extension otherwise an error will occur.

Data Type

String

Using the Magnifcation Property

```
' Display the image at 50% of actual size
ImageMan1.Magnification = 50

' Do the refresh to redraw the Image
ImageMan1.Refresh
```

Mirror Property

Description
Setting this property causes the image to be mirrored vertically or horizontally.

Usage
ImageControl.**Mirror** = 0 | 1

Remarks
The Mirror property settings are:

Setting	Description
HORIZ (0)	The image will be mirrored horizontally.
VERT(1)	The image will be mirrored vertically.

Note
This property is write-only and available only at runtime. Each time the property is set to a value the specified transformation will take place.

Data Type
Integer(Enumerated)

MouseDown Event

Description
Occurs when the user presses a mouse button.

Syntax
Sub ctlname_MouseDown([Index as Integer,] Button as Integer, Shift as Integer, X as Single, Y as Single)

Remarks
MouseDown uses these arguments:

Argument	Description
Index	Uniquely identifies a control in a control array.
Button	Identifies which button was pressed. The Button argument is a bit field with bits corresponding to the left button (bit 0), right button (bit 1), and the middle button (bit 2) - values 1,2,4, respectively. Only one bit will be set, indicating which button caused the event.
Shift	The state of the Shift and Ctrl keys when the button was pressed. The Shift argument is a bit field with bits corresponding to the SHIFT key(bit 0), and the CTRL Key(bit 1) - values 1,2 respectively. One or both of the bits can be set indicating that one or both of the keys was pressed.X,Y The current location of the mouse pointer. X and Y are always expressed in terms of the coordinate system set by the ScaleHeight, ScaleWidth, ScaleLeft and ScaleTop properties of the object.

Note
If the Zoom property is set to TRUE(-1) then no events will be generated for the left mouse button.

MouseMove Event

Description
Occurs when the user moves the mouse.

Syntax
Sub ctlname_MouseMove([Index as Integer,] Button as Integer, Shift as Integer, X as Single, Y as Single)

Remarks
MouseMove uses these arguments:

Argument	Description
Index	Uniquely identifies a control in a control array.
Button	The state of the mouse buttons, in which a bit is set if the button is down. The Button argument is a bit field with bits corresponding to the left button (bit 0), right button (bit 1), and the middle button (bit 2) - values 1,2,4, respectively. All or some or none of these bits can be set.
Shift	The state of the Shift and Ctrl keys. The Shift argument is a bit field with bits corresponding to the SHIFT key(bit 0), and the CTRL Key(bit 1) - values 1,2 respectively. One or both of the bits can be set indicating that one or both of the keys was pressed
.X,Y	The current location of the mouse pointer. X and Y are always expressed in terms of the coordinate system set by the ScaleHeight, ScaleWidth, ScaleLeft and ScaleTop properties of the object.

Note
If the Zoom property is set to TRUE(-1) then no events will be generated for the left mouse button.

MouseUp Event

Description
Occurs when the user releases a mouse button

Syntax
Sub ctlname_MouseUp([Index as Integer,] Button as Integer, Shift as Integer, X as Single, Y as Single)

Remarks
MouseDown uses these arguments:

Argument	Description
Index	Uniquely identifies a control in a control array.
Button	Identifies which button was released. The Button argument is a bit field with bits corresponding to the left button(bit 0), right button (bit 1), and the middle button (bit 2) - values 1,2,4, respectively. Only one bit will be set, indicating which button caused the event.
Shift	The state of the Shift and Ctrl keys when the button was released. The Shift argument is a bit field with bits corresponding to the SHIFT key(bit 0), and the CTRL Key(bit 1) - values 1,2 respectively. One or both of the bits can be set indicating that one or both of the keys was pressed.
X,Y	The current location of the mouse pointer. X and Y are always expressed in terms of the coordinate system set by the ScaleHeight, ScaleWidth, ScaleLeft and ScaleTop properties of the object.

Note
If the Zoom property is set to TRUE(-1) then no events will be generated for the left mouse button.

Overwrite Property

Description

Usage
ImageControl.**Overwrite**[= setting%]

Remarks
The Overwrite property settings are:

Setting	Description
0	Do not overwrite - image will not be saved.
1	Overwrite the existing file.
2	Display a messagebox to allow the user to cancel the save.

Note
If saving to a TIFF or DCX file with the Append property set to TRUE this property will be ignored.

Data Type
Integer(Enumerated)

Pages Property

Description
Returns the number of images/pages in the current file.

Usage
ImageControl.**Pages**

Remarks
This property is read-only. Currently only TIFF and DCX image formats support multiple images.

Data Type
Integer (Read-Only)

PageNumber Property

Description
Sets or Returns the number of the image in the current file.

Usage
ImageControl.**PageNumber** [= num%]

Remarks
The allowable values for this property are zero to *Control.Pages* - 1.

Data Type
Integer

PaletteEntries Property

Example

Description

Returns the number of entries in an image's palette.This property is read-only.

Usage

ImageControl.**PaletteEntries**

Remarks

Use this property with the Red, Green and Blue properties to get or set the colors for an image.

Data Type

Integer

Example

This example prints an images palette

```
Sub DisplayColors()
        Dim I%
        Print "Image Colors"
        Print
        For I=0 To Img.PaletteEntries
                Print Red(I), Green(I), Blue(I)
        Next I
End Sub
```

Picture Property

Example

Description

Specifies the name of the image file to be displayed in the control.

Usage

ImageControl.**Picture**[=picname$]

Remarks

This property specifies the filename of the image to be displayed. If this property is set to a new value which is not an existing file, an error will occur. To clear a picture from the screen set this property to an empty string.

Note

Unlike the Visual Basic Picturebox, images are not stored in the form file.

Data Type

String

Picture Property Example Code

```
' Load a specific image by setting the Picture property to its name
ImageMan1.Picture = "c:\test.tif"

' Clear the current image from the image control
Imageman1.Picture = ""
```

PrnHdc Property

Example

Description
Specifies the printer device context used for printing images.

Usage
ImageControl.**PrnHdc** = Printer.hdc

Remarks
This property must be set to the value of the Printer display context before printing an image. To do this set PrnHdc equal to Printer.hdc.

Data Type
Integer

Note
Failure to set this property before printing an image can result in Unrecoverable Application errors.

Quality Property

Description
Specifies the quality factor used when saving JPEG images.

Usage
ImageControl.**Quality**[= setting%]

Note
The range of legal values for this property are 0 (best compression - worst quality) to 99 (worst compression - best quality).

Data Type
Integer

Paint Event

Description
Occurs when the image in the control has been repainted.

Syntax
Sub ctlname_Paint([Index as Integer,])

Remarks
This control can be used to draw objects on top of the image control and to redraw those items when the image control has been repainted.

Red Property

Example

Description

Sets the red value for a color entry in the image's palette.

Usage

ImageControl.**Red**(Index%)[= Setting%]

Remarks

Use this property along with the Green and Blue properties to change the colors in a 256 color image. After changing these properties you must call the Refresh method to redraw the image with the new colors. The Index% value must be between 0 and the value of the PaletteEntries property. The allowable values for the property are 0 to 255.

Data Type

Integer Array

ReduceTo Property

Description
Setting this property causes the image to be converted to an image of the specified color format.

Usage
ImageControl.**ReduceTo** = 1 | 2 | 3 | 4 | 5 | 6

Remarks
The ReduceTo property settings are:

Setting	Description
B&W (1)	The image will be converted to a black & white (1 bit per pixel) image.
GRAY(2)	The image will be converted to an 8 bit grayscale image.
16COLOR(3)	The image will be converted to a 16 Color image.
256COLOR(4)	The image will be converted to a 256 Color image.
16FIXED(5)	The image will be convertet to a 16 color image with a fixed palette.
256FIXED(6)	The image will be converted to a 256 color image with a fixed palette.

Note
This property is write-only and available only at runtime. Each time the property is set to a value the specified transformation will take place. Currently the color format can only be reduced ie from 24 bit to 8 bit, the control doesn't support increasing the color depth.

Data Type
Integer(Enumerated)

Rotate Property

Example

Description
Setting this property causes the image to be rotated to the specified angle.

Usage
ImageControl.**Rotate** = Angle%

Remarks
You may specify any angle between 0 and 360.

Note
This property is available only at runtime. Each time the property is set to a value the specified transformation will take place.
Reading this property will return the total number of degrees the image has been rotated since being loaded.

Data Type
Integer

Rotate Property Example Code

```
' Rotate an image by 45 degrees

ImageMan1.Rotate = 45
If ImageMan1.ErrCode > 0 then
     MsgBox "Error Occurred during rotation"
End If

' Rotate the image by another 45 degrees
ImageMan1.Rotate =   45

' Since rotation is cummulative, reading the Rotation property should
return 90 degrees
Print ImageMan1.Rotation
```

SaveAs Property

Example

Description

Saves the current image or the specified portion into a file of the specified name and type.

Usage

ImageControl.**SaveAs** = filename$

Remarks

Setting this property will cause the file to be written. If an empty string is specified, a File Save dialog will be displayed so the user can select a file type and name.

If the property is set to a filename, the name specified must contain a supported image extension since the extension is used to determine which format the image is saved in.

To save only a portion of an image use the SrcTop, SrcLeft, SrcRight and SrcBottom properties to specify the portion.

Data Type

String

SaveOptions Property

Example

Description
Specifies the options to be used when saving an image using the SaveAs property.

Usage
ImageControl.**SaveOptions** = options$

Remarks

The following options can be specified using the SaveOptions property:

TIFF_COMPRESS *GROUP3 | MODCCITT | PACKBITS |GROUP4*
Specifies the compression to be used when saving Black and White TIFF images.

TIFF_ROWSPERSTRIP
Specified the number of image rows per TIFF strip in the resulting image. Most applications will not need to manipulate this parameter.

Data Type
String

Saving Images using the SaveAs and SaveOptions Properties

```
' Write the image as a TIFF file with only 1 strip of data
ImageMan1.SaveOptions = "TIFF_ROWSPERSTRIP = "+STR$
(ImageMan1.ImageHeight)

' Save the Image, displaying a SaveAs dialog
ImageMan1.SaveAs = ""
```

ScaleHeight, ScaleWidth Properties

Description
Sets or returns the range of the vertical (ScaleHeight) and horizontal (ScaleWidth) axis for a control's internal coordinate system. The coordinate system is used for scaling and displaying the image in the control.

Usage
ImageControl.**ScaleHeight**[= scale!]
ImageControl.**ScaleWidth**[=scale!]

Remarks
By default, these properties are set to the width and height of an image in ScaleMode 0. If the ScaleWidth and/or ScaleHeight properties are set to less than the ImageWidth and/or ImageHeight properties then only a portion of the image will be displayed in the control. The ScaleLeft and ScaleTop properties can be used to scroll the image so another portion is displayed.

Note
All mouse, scroll, and zoom events generated by the control will have coordinates that are relative to the scaling mode in effect.

Data Type
Single

ScaleHeight, ScaleWidth Properties

Description
Sets or returns the range of the vertical (ScaleHeight) and horizontal (ScaleWidth) axis for a control's internal coordinate system. The coordinate system is used for scaling and displaying the image in the control.

Usage
ImageControl.**ScaleHeight**[= scale!]
ImageControl.**ScaleWidth**[=scale!]

Remarks
By default, these properties are set to the width and height of an image in ScaleMode 0. If the ScaleWidth and/or ScaleHeight properties are set to less than the ImageWidth and/or ImageHeight properties then only a portion of the image will be displayed in the control. The ScaleLeft and ScaleTop properties can be used to scroll the image so another portion is displayed.

Note
All mouse, scroll, and zoom events generated by the control will have coordinates that are relative to the scaling mode in effect.

Data Type
Single

ScaleLeft, ScaleTop Property

Description
Sets or returns the horizontal (ScaleLeft) and vertical (ScaleTop) coordinates that describe the left and top corners of the control's internal area.

Usage
ImageControl.**ScaleLeft**[= scale!]
ImageControl.**ScaleTop**[=scale!]

Remarks
By default, these properties are set to 0. Modifying these properties will affect which portion of the image is displayed in the control. Changing these coordinates will cause the image to be scrolled in the control. These coordinates are automatically updated when the control has the scrollbar property enabled and scrollbars are displayed.

Data Type
Single

ScaleLeft, ScaleTop Property

Description
Sets or returns the horizontal (ScaleLeft) and vertical (ScaleTop) coordinates that describe the left and top corners of the control's internal area.

Usage
ImageControl.**ScaleLeft**[= scale!]
ImageControl.**ScaleTop**[=scale!]

Remarks
By default, these properties are set to 0. Modifying these properties will affect which portion of the image is displayed in the control. Changing these coordinates will cause the image to be scrolled in the control. These coordinates are automatically updated when the control has the scrollbar property enabled and scrollbars are displayed.

Data Type
Single

ScaleMethod Property

Description
Specifies the method ImageMan/VBX will use when scaling images.

Usage
ImageControl.**ScaleMethod**[= setting%]

Remarks
The ScaleMethod property settings are:

Setting	Description
0	Delete bits when scaling.
1	AND bits
2	OR bits
3	Use AntiAliasing when scaling

Note

Data Type
Integer(Enumerated)

Scroll Event

Description
Occurs when the user scrolls the image using the built-in scrollbars.

Syntax
Sub ctlname_Scroll([Index as Integer,] ScaleLeft as Single, ScaleTop as Single)

Remarks
Scroll uses these arguments:

Argument	Description
Index	Uniquely identifies a control in a control array.
ScaleLeft	The new value of the ScaleLeft property.
ScaleTop	The new value ScaleTop property.

Scrollbars Property

Description
Determines whether scrollbars will be displayed when only a portion of the image is displayed in the control.

Usage
ImageControl.**Scrollbars**[= Bool%]

Remarks
The Scrollbars property settings are:

Setting	Description
True(-1)	Enables the scrollbars when the image is scaled so that it is larger than the control. The user can scroll the image by manipulating the scrollbars. When the user scrolls the image your application will receive a Scroll event.
False(0)	Disables scrollbars.

Data Type
Integer(Boolean)

SrcLeft, SrcTop Properties

Description
Specifies the coordinates of the upper left hand corner of the image which should appear in the bounding rectangle when printing an image.

Usage
ImageControl.**SrcLeft**[= left!]
ImageControl.**SrcTop**[= top!]

Remarks
These properties define what portion of the image is scaled to fit in the bounding rectangle defined by the Dst family of properties. By default, the SrcLeft and SrcTop properties are set to 0. SrcLeft must be set to a value between 0 and ImageWidth-1, while SrcTop must be set to a value between 0 and ImageHeight-1.

Data Type
Single

Note
If you want to change these values, you must change them before setting the DoPrint property to print the image.

SrcRight, SrcBottom Property

Description
Specifies the coordinates of the lower right hand corner of the image which should appear in the bounding rectangle when printing an image.

Usage
ImageControl.**SrcRight**[= right!]
ImageControl.**SrcBottom**[= bottom!Remarks]

These properties define what portion of the image is scaled to fit in the bounding rectangle defined by the Dst family of properties. By default, the SrcRight and SrcBottom properties are set to ImageWidth-1 and ImageHeight-1, respectively. SrcRight must be set to a value between 0 and ImageWidth-1, while SrcBottom must be set to a value between 0 and ImageHeight-1.

Data Type
Single

Note
If you want to change these values, you must change them before setting the DoPrint property to print the image.

Select Event

Example

Description

Occurs when the user selects a portion of the image by clicking and dragging and the Select property is set to TRUE(-1).

Syntax

Sub ctlname_Zoom([Index as Integer,] X1 as Single, Y1 as Single , X2 as Single, Y2 as Single)

Remarks

Select uses these arguments:

Argument	Description
Index	Uniquely identifies a control in a control array.
X1,Y1	The coordinates of the upper left corner of the users selection. X1 and Y1 are always expressed in terms of the coordinate system set by the ScaleHeight, ScaleWidth, ScaleLeft and ScaleTop properties of the object.
X2,Y2	The coordinates of the bottom right corner of the users selection. X1 and Y1 are always expressed in terms of the coordinate system set by the ScaleHeight, ScaleWidth, ScaleLeft and ScaleTop properties of the object.

Status Event

Description
Occurs when loading or saving an image.

Syntax
Sub ctlname_Status([Index as Integer,] Percent As Integer)

Remarks
Status uses these arguments:

Argument	Description
Index	Uniquely identifies a control in a control array.
Percentage	Specifies the percentage of completion of the operation.

UseDDB Property

Description
Determines whether the control will display an image using a Device Dependent Bitmap (DDB) or as a Device Independent Bitmap (DIB).

Usage
ImageControl.**UseDDB**[= Bool%]

Remarks
The default setting for this property is True(-1) which enables the use of DDBs.

Using DDBs will speed screen redraw speed while taking more memory. On machines with small memory configurations setting this property to False will conserve memory.

Data Type
Integer(Boolean)

Select Property

Example

Description

Determines whether the control will display a rubber box when the user clicks and drags using the left mouse button. When the user releases the button a Select event will be generated.

Usage

ImageControl.**Select**[= Bool%]

Remarks

The Select property settings are:

Setting	Description
True(-1)	The control will display a rubber box when the user clicks and drags the cursor with the left mouse button.
False(0)	All left mouse button activity will generate MouseDown, MouseMove and MouseUp events.

Data Type

Integer(Boolean)

Using the Select Event & Property

```
' Enabled the user to select an area of the image and copy the
selected
' portion onto the clipboard. The Select property must be set to True.

Sub ImageMan1_Select (X1 As Single, Y1 As Single, X2 As Single, Y2 As
Single)

      ' Set the Src properties to the portion the user selected
      ImageMan1.SrcLeft = X1
      ImageMan1.SrcTop = Y1
      ImageMan1.SrcRight = X2
      ImageMan1.SrcBottom = Y2

      ' Copy  the selected portion to the clipboard
      ImageMan1.ClipboardCommand = 1

End Sub
```

hPalette Property

Description
Returns a handle to a Windows Palette Object containing the palette for the image.

Usage
ImageControl.**hPalette**

Data Type
Integer

VBPicture Property

Example

Description
Sets or Returns the handle to an image in a format compatible with the Visual Basic Picturebox 'Picture' property.

Usage
ImageControl.**VBPicture**[= Picture]

Remarks

This property allows images to be moved between the ImageMan image control and the Visual Basic Picturebox and image controls.

Data Type
Integer

Using VBPicture Property

```
' Load an image into an ImageMan Control and then copy it to a VB
Picturebox

ImageMan1.Picture = "sample.tif"
Picture1.Picture = ImageMan1.VBPicture

' Copy an image from a Picture Control to the ImageMan image Control
Picture1.Picture = LoadPicture("sample.bmp")
ImageMan1.VBPicture = Picture1.Picture
ImageMan1.Refresh
```

ImageMan/VBX Version 5.00
Supported Image Formats

TIFF

PCX

DCX

GIF

BMP

DIB

WMF

EPS

Targa

IMG

JPG

WPG

DXF

PCD

PNG

AutoCad DXF

CompuServe PNG

Kodak Photo CD

TIFF (Tag Image File Format)

ImageMan/VBX supports single and multi page TIFF 5.0 files including the following compression schemes:

No Compression
Packbits
LZW *
Modified CCITT
CCITT Group 3 1d & 2d
CCITT Group 4

Supported color formats include:

Monochrome
16 Color
256 Color
24 Bit Color

*Requires license from Unisys Corp. to use in applications. By default this compression is disabled in ImageMan/VBX. A Unisys license is required to enable support. Click here for information on contacting Unisys Corp.

PCX

Supported color formats include:

- Monochrome
- 16 Color
- 256 Color
- 24 Bit Color

Windows Bitmap (DIB)

ImageMan/VBX supports the following color formats:

Monochrome
16 Color
256 Color
24 Bit Color
16 Color RLE Compressed
256 Color RLE Compressed

OS/2 formatted Bitmap files are also supported.

GIF*

ImageMan supports all GIF87 and GIF89a non-interlaced files with up to 256 colors.

*Requires license from Unisys Corp. to use in applications. By default this compression is disabled in ImageMan/VBX. A Unisys license is required to enable support. Click here for information on contacting Unisys Corp.

WMF (Windows Metafile)

ImageMan/VBX supports all placeable Metafiles.

Targa

ImageMan/VBX supports all version 1 and version 2 targa files with or without compression in the following color formats:

8 bit Greyscale
8 Bit Color
15/16/24/32 Bit color

EPS

ImageMan/VBX supports all encapsulated Postscript files.

IMG (Gem Image Format)

ImageMan/VBX supports compressed, monochrome IMG files.

Changes from ImageMan/VBX Version 3.xx

Image Control

- Support for new image formats including AutoCad DXF, CompuServe PNG and Kodak Photo CD
- Ability to read & write compressed images to database when bound to a data control. Using the MemoryFormat property you can specify what image format and compression is used when saving image data to a database field.
- The hImageData, ImageDataSize and MemoryFormat properties allow you to read and write image data from memory.
- Added the DisplayColors property which returns the number of colors the current video mode supports.
- Faster Color Reduction & Dithering code.
- Added the ScaleMethod property which controls the method used when reducing an image's size.
- Added the VBPicture property. This property is compatible with the VB picturebox's Picture property. You can now move images between ImageMan and the Picturebox by assigning the VBPicture and Picture properties.
- Added Invert property to to invert image colors. Set this to a value of 1 to invert the image. This property is Write-Only.
- Loading an image can now be cancelled by setting the Percentage parameter of the Status event to -1.
- Added the GotFocus and Lostfocus events.
- Added a Paint Event. This event is fired after the control is painted.
- The File Open & File Save dialogs are now centered when displayed.
- The Rotate property now supports single degree rotation and is now read/write. Reading the property returns the cummulative rotation for an image since it was loaded.
- The JPEG writer now supports writing images of any color depth. The writer will internally convert the image data to 24 bits.
- Added the AppendImage, Compression, Overwrite and Quality properties for simplifying saving images. Now most image parameters can be specified using these properties instead of the dreaded SaveOptions property. See the online help for more info on these properties.
- Added the IncreaseTo property to allow the color depth of an image to increased.

- Fixed a bug which returned an error when the control was bound to an empty recordset.
- Fixed a bug which called a VB 2.0 API function when using Visual C++ or any other non Visual Basic host environment.
- Fixed a bug which caused the palette not to be updated when changing an image's gamma.
- Fixed a bug when copying Metafiles from the Windows Clipboard into the image control
- Misc. other bug fixes.

Scanner Control

- Added Brightness and Contrast properties to allow these values to be programmatically adjusted.
- Fixed a bug which caused the image control to sometimes leave temporary files around.
- Cancelling during a multi-page scan now stops the scan process.

JPG (JPEG Image Format)

ImageMan/VBX supports JFIF standard JPEG files.

WPG (WordPerfect Graphics Format)

ImageMan/VBX supports both raster and vector WPG files.

ClipboardCommand Property Example

```
' Copy an Image to the Clipboard
ImageMan1.ClipboardCommand = 1

' Copy an Image from the Clipboard to the control
If Clipboard.GetFormat( CF_DIB ) Then
      ImageMan1.ClipboardCommand = 2
      ImageMan1.Refresh
End If
```

ImageMan/VBX Online Help
Version 5.00
1 June 1996

Contacting Unisys Corporation

The LZW algorythm used in GIF and some TIFF files is subject to a patent issued to Unisys Corporation.

To be able to unlock the GIF reader/writer and the TIFF modules with LZW support you must obtain a license from Unisys. Once you have the agreement you must fax the first page to DTI and we will respond with the password to unlock the IMLZW50.ZIP file which contains the LZW enabled files.

Unisys Corporation
Welch Licensing Dept. - C1SW19
PO Box 500
Blue Bell, PA 19424

Voice: 215-986-4411
Fax: 215-986-5721