Disclaimer of Warranties and Limitation of Liabilities

Contents

Professional Developer's Guide

1 Imaging In Development

Welcome to the Professional Developer's Guide 2

Introducing Imaging for Windows 3

Imaging Components 4

Imaging Application 4

Imaging Preview 5

Imaging Flow 6

What Exactly Is Document Imaging? 20
Business Document Imaging 20

The Automation From Excel Project 40
Opening the Spreadsheet File 42

Annotation Selection 118
Example 119
Copy Image Project 119
Printing An Image 122
Image Printing Defined 122
Example 122
Print Image Project 124
Scanning an Image Using a Template 130
Template Scanning Defined 130
Example 131
Template Scan Project 133
Managing an Image File Using Thumbnails 143
Thumbnails Defined 143
Example 143
Thumbnail Sorter Project 144
Subtracting the Value of X 156
Unloading a Multipage Image File 158

The following sections explain each server option setting and related property in detail. 177
File Location for Document Pages (FileStgLoc1x Property) 177
Force Lower-Case File Names (ForceLowerCase1x Property) 178
Link Files On Reference (ForceFileLinking1x Property) 178
Delete Files With Pages (ForceFileDeletion1x Property) 180
Browsing for Volumes or Image Files and Server Documents 181
Browsing for Volumes 181
Browsing for Files and Documents 182
Querying for 1.x Server Documents 185

AnnotationLineColor Property 289
AnnotationLineColor Example — VB 290

Contents

Contents

Hollow Rectangle 533
Text 533
Text From File 534
Freehand Line 534

Contents

Imaging Flow Samples 861
Flow Program 861
Flow Variables 861

B Imaging ActiveX Tips and Tricks

Tips and Tricks 864
Miscellaneous Programming Tips 864
How to use functions of the *Version 2.0* **ActiveX sample applications 864**
Specify *tenths*

Optical Character Recognition Tips 873
When working with Interactive Training 873
When performing OCR 874

C Imaging ActiveX Controls Summary

Overview 878
Imaging ActiveX Table Definitions 878

Development Tools and Methods

When end users

OLE

OLE lets you add a subset of Imaging functions to your application. It is

At flow design time, the author can set Review tool options that manipulate the Imaging application as well as the image it displays. These

Sample Code

The companion CD contains sample code designed to help you add Imaging functions to your applications.

- Copy an image.
- Print an image.
- Scan images using a template.
-

The Imaging for Windows 95, 98, and NT controls are free; as such,

Chapter 2

Embedded Server Mode

To create the Application and Image File Objects

1. Declare the object variables that will contain references to the Application and Image File objects.
2. Use the **Set** statement and the **CreateObject** function of Visual Basic to create and return a reference to the Application object.
3. Use the **Set** statement of Visual Basic and the **CreateImageViewerObject** method of the Application object to create and return a reference to the ImageFile object.

With the Application and ImageFile objects instantiated, you can

Chapter 2

Demonstration Project

This section demonstrates how to automate the Imaging application from Microsoft Excel.

While a wide-ranging discussion of every Imaging function is beyond the scope of this chapter, the information presented here is sufficient to get started.

The demonstration project was developed using Microsoft Visual Basic for Applications, Version 5.0 and Excel 97.

Thumbnail

The Thumbnail view mode lets users display image files as a series of thumbnail images — one for each image page. It lets users:

- View multiple image pages simultaneously.
- Rearrange pages using drag and drop.
- Delete pages.
- Drag and drop pages to and from other applications that support drag and drop functionality.

Rearranging image pages is available with Imaging for Windows Professional Edition only.

Keep in mind that some Imaging functions — like annotation and zoom — are not available in this mode because they are not appropriate for use on such small images.

Opening the Spreadsheet File

Start Excel and then open the `ImagingAutomation.xls` file. The sample spreadsheet appears.

Opening and Displaying the Image File

Give focus to Cell A1, which contains the path and file name of the sample TIFF image file.

Rotating an Image Page

On the **Tools** menu in Excel, point to **Macro**

3

Automation Lexicon

This chapter describes the properties and methods of each Imaging for Windows Automation object.

In This Chapter

Overview 52

Overview

This chapter describes the properties and methods of each object in the Imaging application object hierarchy.

Application Object

The Application object is a top-level object that controls every other

Remarks This property must be set prior to opening the ImageFile object. It only takes effect if the **Width**,

The **ImageView** property and the **ImageFileObject.ActivePage** property have the following relationships:

See Also ImageFileObject.**ActivePage** property.

ImagingToolBarVisible Property

Chapter 3

Application Object Methods

The following table lists the Application object methods.

CreateImageViewerObject Method

PageCount Property

Description

ImageFile Object Methods

The following table lists the ImageFile object methods.

ImageFile Object Methods

Method	Description
AppendExistingPages	Appends existing pages to the end of the ImageFile object.
Close	Closes the ImageFile object.
CreateContactSheet	Saves a contact sheet rendition of the ImageFile object. Professional Edition only.
FindOIServerDoc	Finds *Eastman Software* Imaging Server documents. Not available when the application is running as an embedded server. Professional Edition only.
Help	Displays online Help.
InsertExistingPages	Inserts existing pages in the ImageFile object.
New	Creates a new blank ImageFile object. Not available when the

Ocr Method

Description OCRs all image file pages.

Usage *ImageFileObject*.**Ocr**

Remarks The Image file must be open. The **Ocr** method uses the **OcrOutputFile** and **OcrOutputFileType** properties.

Example

```
'This example performs an OCR on an image object.
Dim App, Img As Object
Set App = CreateObject("Imaging.Application")
Set Img = App.CreateImageViewerObject(1)
Img.Open "d:\pcx.tif"
Img.Ocr
```

Open Method

Description Opens an image file in the parent application window. This associates an image file with the ImageFile object. If a file is currently open,e.arr.

Ocr Method

Description OCRs the page range.

UsageUsage *PageRangeObject*.**Ocr**( )

Example

```
'This example OCRs pages 2 through 6.
x = Img.Pages(2,6).Ocr
```

Peı˝f67 -1.895j3nge.

4

Visual Basic

To add the Imaging ActiveX controls to Visual Basic

1. Start Visual Basic and create a new project.
2. On the **Project** menu, click **Components**.
3. On the **Components** dialog box, click the

Access

7 In the **Categories** list box:

a Click **ActiveX Controls**.

b From the **Commands** list box, drag each Imaging ActiveX control and drop it onto the **Controls** toolbox.

Commands

c When you finish dragging and dropping the Imaging ActiveX controls onto the toolbox, click **Close**

Obtaining Help

This section explains how to access the on-line help system of the Imaging

Chapter 4

Visual C++

There are two ways to access the Imaging ActiveX Controls on-line help in Visual C++. You can access help from the:

- **Components and Controls Gallery** dialog box
- **Properties** window

4 Click the desired property in the **Properties**

256 Colors

Example

Example

Users of your application may want to change the resolution of an image

Convert Image Project

The Convert Image project shows how to provide image file type and page property conversion functions to your users.

The project consists of one form and the following controls:

- One Image Admin control
- One Image Edit control
-

Chapter 4

Finally, the procedure obtains the current image content of the Clipboard using Visual Basic's **GetData** method and displays it in the PictureBox

Print Image Project

The Print Image project demonstrates printing an image file.

The project consists of one form and the following controls:

- One Image Admin control
- One Image Edit control
- Two Command button controls in a control array

And it uses the following Imaging methods to provide the print image function:

PrintRangeOption The following

```
Private Sub cmdPrint_Click(Index As Integer)

    Select Case Index

        Case 0 'Print

            On Error GoTo Print_EH

            'Display the Print dialog box.
            kdkImgAdmin1.ShowPrintDialog frmPrintImage.hWnd

            'User pressed OK continue with print
            If kdkImgAdmin1.StatusCode = 0 Then
```

```
Private Sub cmdOK_Click()

    'Set the File Type property of the Image Scan
    'control on frmMain according to File Type option
    'button on this form.  In addition, set the Enabled
    'property of the label and text box controls accordingly.
    If optTiff.Value Then
        frmMain!kdkImgScan1.FileType = TIFF
        frmMain!lblPages.Enabled = True
        frmMain!txtPages.Enabled = True
    ElseIf optBMP.Value Then
        frmMain!kdkImgScan1.FileType = BMP_Bitmap
        frmMain!lblPages.Enabled = False
```

- Click **Scan Options** to apply compression. The `mnuCompressionOptions_Click()`

- Click **Stop Button** to have the Imaging software display a **Stop** button while scanning.

The `mnuStopButton_Click()` event in the **Main** form sets the **StopScanBox** property to True or False (as appropriate) to either display, or not display, the **Stop** button. The **Stop**

```
Private Sub cmdStartScan_Click()

    On Error GoTo Scan_EH
.
.
.
    'Link the Image Scan and Image Edit controls to permit display
    'while scanning
    kdkImgScan1.DestImageControl = "kdkImgEdit1"

    'Set the ScanTo property to enable template scanning
    kdkImgScan1.ScanTo = DisplayAndUseFileTemplate

    'Concatenate the path, backslash, and template prefix. Then assign the
    'string to the Image property
```

- Sets the **EnableDragDrop** property of the Image Thumbnail

Chapter 4

Sorting an Image File (Using Drag and Drop)

If necessary, start the Thumbnail Sorter project and select a multipage image file.

Point to the thumbnail of an image page you want to insert before

Chapter 4

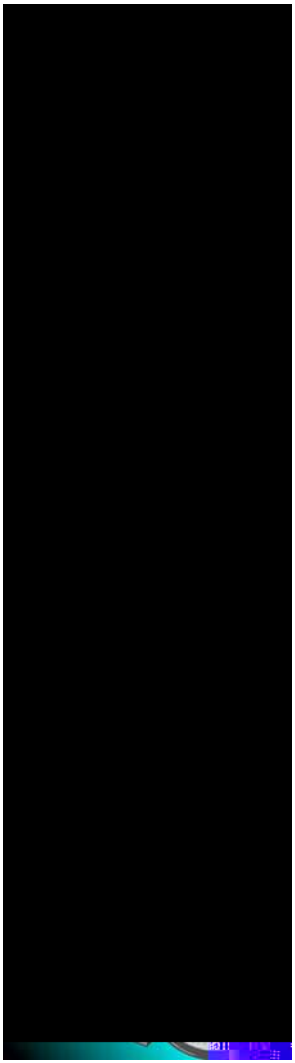
- Invokes the **DeletePages** method of the Image Admin control,

Chapter 4

```
Private Sub cmdUnload_Click()
    Dim intPageCount As Integer, intPageNumber As Integer
    Dim intSlashPos As Integer, intDotPos As Integer
    Dim intVerifyExistence As Integer
    Dim strPrefix As String, strCurrentPath As String
    Dim strSourceFile As String, strUnloadedFileName As String
    Dim strExt As String, strPageNum As String

    'Get the path and file name of the source file
    strSourceFile = kdkImgEdit1.Image
    strPrefix = kdkImgEdit1.Image

    'Get the number of pages in the source file
    intPageCount = kdkImgEdit1.PageCount

    'Get the current path
    strCurrentPath = CurDir

    'Establish an appropriate prefix for the Unloaded file names
    intSlashPos = 7
    'First, eliminate the characters to the left of the slashes
    Do While intSlashPos <> 0
        intSlashPos = InStr(1, strPrefix, "\", 1)
        strPrefix = Right(strPrefix, Len(strPrefix) - intSlashPos)
    Loop
    'Second, eliminate the characters to the right of the dot
    intDotPos = InStr(1, strPrefix, ".", 1)
    strExt = Right(strPrefix, Len(strPrefix) - intDotPos)
    If intDotPos > 4 Then
        strPrefix = Left(strPrefix, 3)
    Else
        strPrefix = Left(strPrefix, intDotPos - 1)
    End If

    'Append a slash to the current path specification
    intSlashPos = InStr(1, strCurrentPath, "\", 1)
    If intSlashPos <> Len(strCurrentPath) Then
        strCurrentPath = strCurrentPath + "\"
    End If
```

5

Developing Client-Server Applications



Imaging Server Concepts

This section explains the basic concepts of Imaging 1.x and Imaging 3.x server access.

Developing applications that can access and interact with *Eastman Software*

File Type Support

The Imaging 1.x and Imaging 3.x servers support the file types described in the following table.

Standard Dialog Boxes

Delete Files With Pages (ForceFileDeletion1x Property)

When users delete a page from an Imaging 1.x server document, they may also want to delete the linked image file.

The **ForceFileDeletion1x** property of the Image Admin control

Browsing for Volumes or Image Files and Server Documents

After users make their selection and click **Open**, the Imaging software

When users select 1.x Files and Documents, they can use the **Look in** list box to browse file and document volumes in the current domain, as illustrated in the following figure.

Chapter 5

GetSelectedAnnotationLineColor method — Returns the line color used in selected Straight Line, Freehand Line, and Hollow Rectangle annotations.

SetSelectedAnnotationLineColor method — Sets the line color to use in selected Straight Line, Freehand Line, and Hollow Rectangle annotations.

GetSelectedAnnotationLineStyle method — Returns the line style

Example

Users of your application may want to annotate image files with important comments. They may also want to link image pages to related Web pages.

Scenario

Kim manages a QA (quality assurance) group in a software company that produces applications for engineer-to-order manufacturing firms around the world. Some of the development and testing of the applications is performed in the United States, the remainder is performed in Ireland.

As part of their jobs, Kim and her staff regularly distribute and peer-review scanned specification and test plan documents. Because the QA group is spread across two continents, each analyst relies on e-mail

Show1xServerOptDlg method — To display the **Imaging Server Options** dialog box, which lets users set Imaging 1.x server options.

ShowFileDialog method — To display the **Open** dialog box, which lets users select the image files or server documents they want to open.

Browse1x method — To display the **Browse 1.x** dialog box, which lets users browse the Imaging 1.x server for server file and/or document volumes.

CreateDirectory method — To create a cabinet, drawer, and/or folder.

ConvertDate method

```
Private Sub mnuFileOpen_Click()

    On Error Resume Next

    '-------------------------------------------------------
```

```
Private Sub cboCabinet_Click()
    Dim objResults As Object
    Dim strSinglePlural As String
    Dim vntItem As Variant
   .
   .
```

```
Private Sub cboFolder_Click()
    Dim objResults As Object
    Dim strSinglePlural As String
    Dim vntItem As Variant
   .
   .
   .
```

On the **1.x Cabinet\Drawer\Folder\Document** window, click the desired document in the **Document** combo box and then click the **Open** button. The `cmdOpen_Click()` event procedure invokes the public subroutine, `PerformFileOpen(strDocManagerCDFD)`, which opens and displays the document selected.

```
Private Sub cmdFind_Click()
    Dim objResults As Object
    Dim vntItem As Variant
    Dim strConvertedDate As String

    lstResults.Clear
    '-----------------------------------------------------------
    ' Perform a query; store Doc names in the lstResults listbox.
    '-----------------------------------------------------------
    If optQuery(0).Value = True Then        'Query by Document
        mstrQuery = "finddocs document " & _
            cboName.List(cboName.ListIndex) & " " & txtName.Text

    ElseIf optQuery(1).Value = True Then    'Query by Date
        strConvertedDate = kdkImgAdmin1.ConvertDate(txtDate.Text)
        mstrQuery = "finddocs " & _
            cboDateName.List(cboDateName.ListIndex) & " " & _
            cboDate.List(cboDate.ListIndex) & " " & strConvertedDate
```

Select a document and click **OK**. The `cmdOK_Click()` event procedure invokes the public subroutine, `PerformFileOpen (kdkImgAdmin1.Image)`

6

Image-Enabling Web Pages

This chapter explains how to use the Imaging ActiveX controls to image-enable your Web pages.

The chapter begins by showing you how to declare and define the Imaging ActiveX controls in a Java applet and HTML source file. It continues by explaining how to access the on-line help for the controls

As an example, the following code snippet shows the proper object reference declarations for two Imaging ActiveX controls: Image Edit and Image Thumbnail.

To connect the Imaging ActiveX controls

1

Obtaining Help

This section explains how to access the on-line help system of the Imaging ActiveX controls.

You can access the Imaging ActiveX Controls on-line help system within Visual J++ or directly from Windows Explorer.

Visual J++

Imaging ActiveX help is accessible from the

Windows Explorer

The following list briefly describes the methods you'll find useful when

Chapter 6

The Imgctrls Project

The **setControl** method saves these references. This action completes the connection between the Imgctls applet and the HTML source code,

The **LoadImage()** method (shown in the following code snippet) uses several properties and methods to display the image document file in the Image Edit and Image Thumbnail controls.

Specifically, it performs the following tasks:

- Sets the **Image**

Chapter 6

7

Image Edit/Image Annotation Controls

This chapter describes the properties, methods, and events for the Image Edit and Image Annotation controls. This chapter also describes prop-

Overview

Image Edit Control

All properties, methods, and events in this chapter apply to the Image Edit control, with the

Image Edit Control Overview

What the Image Edit Control Lets You Do

The Image Edit control lets you — the application developer — add image display, annotation, manipulation, and management functions to applications that support 32-bit ActiveX/OCX controls.

What the Image Edit Control Lets Your Users Do

Depending on how you design and code your application, the Image Edit control lets your users display, annotate, and manipulate image document files and managed image documents.

Image Display

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

Image Annotation

The control includes several properties, methods, and events that enable you to add annotation functions to application programs. The annotation functions permit your end users to annotate displayed images.

You can set the annotation types [anns.

Image Manipulation

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

- Convert to text (when used in conjunction with the Image OCR control)
- Copy, cut, and paste to the Clipboard
- Crop
- Deskew
- Despeckle
- Flip
- Invert
- Rotate
-

Establishing the Link

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

What the IATB Control Lets Your Users Do

Each IATB control lets your users draw text or graphical annotations on displayed image documents.

Prerequisites

You must include at least one Image Admin control and one Image Edit control in your application.

Further, you must link each IATB control to the Image Edit control that displays images to be annotated by your end users.

The link permits the IATB control to send messages to the Image Edit control. These messages change the annotation properties of the Image Edit control to the values set

```
    'Sets the AnnotationType to a filled rectangle and the fill color of the
    'rectangle to blue using Visual Basic color constants.
    ImgAnnTool1.AnnotationType = wiFilledRect '4
    ImgAnnTool1.AnnotationFillColor = vbBlue
    'Draw the blue rectangle on the Image Edit control whose ImageControl
    'name is ImgEdit1
    ImgAnnTool1.DestImageControl = "ImgEdit1"
    ImgAnnTool1.Draw 10, 10, 100, 100
End Sub
```

AnnotationFillStyle Property

Data Type IFontDisp.

Remarks

Example 1

```
Private Sub cmdStamp_Click()
    ImgEdit1.AnnotationFont = "Courier"
    ImgEdit1.AnnotationFontColor = vbRed
    'This stamp uses embedded text macros for date and time
```

AnnotationImage Property

Description

AnnotationLineStyle Example — VC++

This example draws a transparent line across the width of an image.

```
void CImgEdit1Dlg::OnDrawline()
{
    // This would draw a line across the width of an image
    long XWidth;
    // Determine the width of e th/˝4
```

See Also AnnotationLineStyle property, AnnotationType property, GetSelectedAnnotationLineWidth method, SetSelectedAnnotationLineWidth method.

AnnotationLineWidth Example — VB

This example draws a 10-pixel-wide line across the width of an image.

```
Private Sub cmdDrawLine_Click()
    Dim XWidth As OLE_XSIZE_PIXELS
    'Determine the width of the image -- this will work even if the image
    'is scaled up or down
    XWidth = ImgEdit1.ImageScaleWidth
    'line will be green, transparent, 10 pixels wide
    ImgEdit1.AnnotationLineColor = vbGreen
    ImgEdit1.AnnotationLineStyle = wiTransparent  '0
    ImgEdit1.AnnotationLineWidth = 10
    ImgEdit1.AnnotationType = wiStraightLine  '1
    'Draw the line
    'left position = 0 (start at left side)
    'top position = 100 pixels from top of page
    'width of line = width of image as it is displayed
    'height of line = 0 (line will be horizontal)
    ImgEdit1.Draw 0, 100, XWidth, 0
End Sub
```

AnnotationLineWidth Example — VC++

This example draws a 10-pixel-wide line across the width of an image.

```
void CImgEdit1Dlg::OnDrawline()
{
    // This would draw a line across the width of an image
    long XWidth;
    // Determine the width of the image -- this will work even if the image
    // is scaled up or down
    XWidth = ImgEdit1.GetImageScaleWidth();
    // line will be green, transparent, 10 pixels wide
    ImgEdit1.SetAnnotationLineColor(0xFF00);    // vbGreen
    ImgEdit1.SetAnnotationLineStyle(0);  // 0
    ImgEdit1.SetAnnotationLineWidth(10); // 10
    ImgEdit1.SetAnnotationType(1); // wiStraightLine 1
    // Draw the line
    // left position = 0 (start at left side)
    // top position = 100 pixels from top of page
    // width of line = width of image as it is displayed
    // height of line = 0 (line will be horizontal)
    VARIANT vWidth; V_VT(&vWidth) = VT_I4; V_I4(&vWidth) = XWidth;
    VARIANT vHeight; V_VT(&vHeight) = VT_I4; V_I4(&vHeight) = 0;
    ImgEdit1.Draw(0, 100, vWidth, vHeight);
}
```

AnnotationOcrType Example — VB

The stamp text can include the following text macros:

See Also

Data Type String.

Remarks The name must be a fully-qualified file name. A file extension is required for all files.

Applies To Image Edit control.

Usage

```
CompInfo.m_RBO = FALSE;
```

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

Image Property

Description

ImageHeight Example — VB

This example displays an image and scales it so it fits into the window. Then it displays the height, width, resolution, scaled height, and scaled width of the image to illustrate the difference between ImageHeighti5f0g48(e)2he

```
    frmInfo.lblImgScaleHeight.Format("%i",lInfo);
    lInfo = ImgEdit1.GetImageScaleWidth();
    frmInfo.lblImgScaleWidth.Format("%i",lInfo);
    frmInfo.DoModal();
}
```

ImageModified Property

Description Indicates whether an image has been modified.

Available With

Applies To Image Edit control.

Usage *object*.**ImageModified**

Data Type Boolean.

Remarks Because the value of this property indicates whether an image file has been modified, you

Usage *object*.**ImageScaleHeight**

Data Type OLE_YSIZE_PIXELS.

Remarks The **Display**

MagnifierZoom Example — VB

This example scrolls the image to a specified location, sets the Magnifier zoom factor, and then displays the Magnifier window.

Remarks

```
ImgEdit1.ScrollShortcutsEnabled = True
'Display all OCR zones
ImgEdit1.
```

Page Example — VB

```
        ImgEdit1.SetPage(lngNewPg);
        ImgEdit1.Display();
    }
```

PageType Property

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

PageType Example — VB

This example uses the PageType property to determine the page type of the image specified in the Image property.

```
Private Sub cmdPageType_Click()
```

Chapter 7

ReadyState Property

Description

ScrollBars Example — VC++

This example shows some property settings you might want to make prior to displaying an image.

```
void CImgEdit1Dlg::OnLoadEditctrl1(double Zoom)
{
```

See Also AutoRefresh property, Display method, Load event, Scroll event, ScrollBars property, ScrollImage method, ScrollPositionY property.

ScrollPositionX Example — VB

ScrollShortcutsEnabled Example — VB

This example shows some property settings you might want to make prior to displaying an image.

```
Private Sub ImgEdit1_Load(ByVal Zoom As Double)
    'Here are some examples of default settings you might want to specify
    'prior to displaying an image. Repaint any changes immediately
    '(ex. zoom or resolution changes, etc.)
    ImgEdit1.AutoRefresh = True
```

SelectionRectangle Property

Description Returns or sets whether a selection box will be drawn when end users click the left mouse button and drag the mouse pointer over a displayed image.

Available With

Applies To Image Edit control.

Usage

SelectionRectangle Example — VC++

This example draws a selection rectangle on an image and zooms in on the area bound by

```
void CImgEdit1Dlg::OnUndo()
{
    // The Undo method will undo the last imaging operation performed.
```

Zoom Property

Description Returns or sets the zoom factor for an image.

Arguments

```
    ImgEdit1.AutoDeskew
    'Remove speckles on the image using the default
    'threshold value for dot size
    ImgEdit1.Despeckle
    'If there is a lot of extraneous space around image
    'borders, cropping might reduce processing time
    ImgEdit1.AutoCrop
    'Show the OCR dialog box
    ImgOCR1.ShowOCR
End Sub
```

AutoCrop Example — VC++

This example shows how to crop an image automatically. (It also shows how to invoke

Applies To Image Edit control.

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

Arguments The ClipboardCopy method has the following parameters:

Returns None.

Remarks If the ClipboardCopy method is invoked without parameters after an end user draws a

```
        'Color Burn in always converts to RGB24. To save disk space, save the
        'image as an 8 bit palettized image
        ImgEdit1.SaveAs "C:\imgsave\annotate.tif", wiFileTypeTIFF,
        ➥  wiPageTypePal8
    End Sub
```

ConvertPageType Example — VC++

Arguments The Crop method has the following arguments:

Returns None.

Remarks The **Display**

Chapter 7

Arguments The Despeckle method has the following argument:

Returns None.

Remarks

Display Example — VB

This example displays a user-selected image file in an Image Edit control.

```
Private Sub cmdDisplayImage_Click()
    'Use the ImgAdmin Open dialog box to display a file in the Image
    'Edit and Thumbnail controls.
    ImgAdmin1.ShowFileDialog OpenDlg
```

Draw Example — VB

This example uses the SelectTool method to select the Text From File annotation type. It

Arguments The DrawSelectionRect method has the following arguments:

Returns None.

Remarks None.

See Also ClipboardCopy method, ClipboardCut method, DeleteImageData method,

See Also

```
        //Err.Description
    }
    END_CATCH
}
void CImgEdit2Dlg::OnMarkSelect(short Button, short Shift, long Left,
➥ long Top, long Width, long Height, short MarkType, LPCTSTR GroupName)
{
    m_MarkLeft = Left;
```

```
        frmInfo.lblImgScaleHeight.Format("%i",lInfo);
        lInfo = ImgEdit1.GetImageScaleWidth();
        frmInfo.lblImgScaleWidth.Format("%i",lInfo);
        frmInfo.DoModal();
    }
```

GetCurrentAnnotationGroup Method

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

Available With

Applies To

GetRubberStampMenuItems Method

Description

GetSelectedAnnotationFillStyle Method

Description

GetSelectedAnnotationLineStyle Method

```
    Else
      optTransparent.Value = True
    End If
    frmLineStyle.Show
End Sub
```

```
Dim LineColor As OLE_COLOR
'Get the line width and color to initialize the dialog box.
intLineWidth = frmMain.ImgEdit1.GetSelectedAnnotationLineWidth
LineColor = frmMain.ImgEdit1.GetSelectedAnnotationLineColor
```

Returns 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

Returns Long.

```
VARIANT vRotateAmt; V_VT(&vRotateAmt) = VT_I4; V_I4(&vRotateAmt) = 45;
ImgEdit1.RotateRight(vRotateAmt); // optionally can specify degrees
ImgEdit1.Flip(); // 180 degrees
```

RotateRight Example — VB

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

Special Note Regarding Imaging 1.x/3.x Documents and Files

If you want your program to process Imaging 1.x managed documents or files, you must include an Image Admin control in your project and set its **FileStgLoc1x**

Chapter 7

Chapter 7

SelectAnnotationGroup Method

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

Available With

Applies To Image Edit control.

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

Arguments

SetCurrentAnnotationGroup Method

Description Sets the annotation group to which subsequent annotations will belong.

SetRubberStampItem Example — VC++

This example lets users select the type of rubber stamp they want to use.

```
void CImgEdit2Dlg::OnSetcurrentstamp()
{
    // This allows a user to click on a stamp in order to select it as
    // the current stamp. See GetRubberStampMenuItems for an example of how
    // to create a list of stamps.
    // Load the list box.
    CStampList frmStampList;
    frmStampList.m_pParent= this;
    frmStampList.DoModal();
    ImgEdit1.
```

Chapter 7

SetSelectedAnnotationFont Example — VB

This example uses the GetSelectedAnnotationFont method to determine the font attributes

```
COleFont AnnoFont;
```

SetSelectedAnnotationFontColor Example — VC++

This example uses the GetSelectedAnnotationFontColor method to determine the font color of a selected annotation. It then updates the Microsoft common dialog box to indicate the color obtained.

This example also uses the Microsoft common dialog box and the SetSelectedAnnotationFontColor method to change the font color of a selected annotation.

```
void CImgEdit2Dlg::OnAnnofont()
{
    // This example uses the Microsoft font dialog box to
    // change a selected text annotation mark's font.
    COleFont AnnoFont;
    AnnoFont = ImgEdit1.GetSelectedAnnotationFont();
    CY cy;
    long FontColor;
    // Determine the current font used in the selected annotation
    // mark and init the dialog box font property to that font.
```

```
    'mark and init the dialog box color property to that color.
    LineColor = ImgEdit1.GetSelectedAnnotationLineColor
    CommonDialog1.Color = LineColor
    CommonDialog1.Flags = cdlCCRGBInit
    CommonDialog1.ShowColor
    ImgEdit1.SetSelectedAnnotationLineColor CommonDialog1.Color
End Sub
```

SetSelectedAnnotationLineColor Example — VC++

SetSelectedAnnotationLineWidth Method

Description

```
    {
      m_TextZone.SetCheck(0);
      m_PictureZone.SetCheck(1);
    }
    return TRUE;  // return TRUE unless you set the focus to a control
                  // EXCEPTION: OCX Property Pages should return FALSE
}
```

ShowAnnotationGroup Method

Description Shows the specified annotation group.

Available With

ShowAnnotationGroup Example — VB

This example lets a user select an annotation group from a list and make it visible. See the example for the GetAnnotationGroup method for instructions on how to list annotation groups.

```
Private Sub cmdShowGroup_Click()
    Dim strCurGroup As String
    'Determine which group the user selected from the listbox.
    strCurGroup = AnnoGroups.List(AnnoGroups.ListIndex)
    Form1.ImgEdit1.ShowAnnotationGroup (strCurGroup)
End Sub
```

ShowAnnotationGroup Example — VC++

This example lets a user select an annotation group from a list and make it visible. See the example for the GetAnnotationGroup method for instructions on how to list annotation groups.

```
void CFrmGroup::OnShowGroup()
{
    // This would allow a user to select a Group from a list and make it
    // visible. See example for GetAnnotationGroup method for how to list
    // annotation groups.
    CString szCurGroup;
    // Determine which group the user selected from the listbox.
    m_GroupList.GetText(m_GroupList.GetCurSel(),szCurGroup);
    VARIANT vCurGroup;
    V_VT(&vCurGroup) = VT_BSTR;
    V_BSTR(&vCurGroup) = szCurGroup.AllocSysString();
    if(pParentDlg)
    {
      pParentDlg->ImgEdit1.ShowAnnotationGroup(vCurGroup);
    }
}
```

Remarks

ShowAnnotationToolPalette Example — VC++

This example shows how to invoke the annotation tool palette with user-defined tool tips.

Returns None.

Remarks If the position and size parameters are not specified, the magnifier window is shown at its

ShowPageProperties Method

Description

Returns Long.

Remarks

```
'The Undo method will undo the last imaging operation performed.
ImgEdit1.Undo
```

BadDocumentFileType Example — VB

Close Example — VB

Remarks For this event to fire, an image must be displayed and the Image Edit control must be entering or exiting Text Edit mode.

The Image Edit control enters Text Edit mode when:

-

Arguments

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

Arguments The MarkEnd event has the following arguments:

Remarks The MarkType parameter cannot be the Select Annotations annotation type.

Remarks

Chapter 7

ReadyStateChange Event

Description This event occurs when the state of the control has changed. When it fires, check the **ReadyState**

File Types

Imaging ActiveX controls support the following file types:

AWD — The AWD (At Work Document) file type is used by Microsoft Fax. AWD supports the black-and-white page type only; its image documents can contain multiple image pages. AWD image documents are compressed using Microsoft's RBA compression.

8

Image Admin Control

This chapter describes what developers and users can do using the Image Admin control, as well as the properties and methods that are available with this control. In addition to the properties and methods described in this chapter, the following properties and methods, which are described in Chapter 7, apply to the Image Admin control:

- StatusCode property
- AboutBox method
- GetVersion method

In This Chapter

What the Image Admin Control Lets You Do 548

What the Image Admin Control Lets Your Users Do 548

in

in

What the Image Admin Control Lets You Do

The Image Admin control lets you — the application developer — add image file

CancelError Property

Returns or sets whether an error will be returned when a user chooses Cancel from the following dialog boxes: File, Print, LoginToServer, or 1.x Server Options.

Available With

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

Data Type Boolean.

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

```
PrintErr:
    'User pressed the cancel button.
        Exit Sub
End Sub
```

CancelError Example – VC++

This example uses the **ShowPrintDialog**

Comments Example – VC++

```
    'Determine the X and Y resolution of the image page.
    frmInfo.lblImgInfo(4).Caption = ImgAdmin1.ImageResolutionX
    frmInfo.lblImgInfo(5).Caption = ImgAdmin1.ImageResolutionY
    'Determine the number of pages in the file.
    frmInfo.lblImgInfo(7).Caption = ImgAdmin1.PageCount
    'Show the form with the image attributes.
    frmInfo.Show
End Sub
```

```
case 8:
     strImgFileType = "GIF";
           break;
case 9:
     strImgFileType = "WIF";
           break;
}
frmInfo.m_FileType = strImgFileType;
// Read the CompressionType property and translate the value to a
// corresponding string.
switch (ImgAdmin1.GetCompressionType())
{
case 0:
     strImgCompType = "Unknown";
           break;
case 1:
     strImgCompType = "No Compression";
           break;
case 2:
     strImgCompType = "Group3(1D)";
           break;
case 3:
     strImgCompType = "Group3(Modified Huffman)";
           break;
case 4:
     strImgCompType = "PackBits";
           break;
case 5:
     strImgCompType = "Group4(2   break;
case 2:
```

```
// Read the PageType property and translate the value to a
// corresponding string.
switch (ImgAdmin1.GetPageType())
{
```

```
frmInfo.lblImgInfo(3).Caption = ImgAdmin1.ImageWidth
```

```
case 8:
     strImgFileType = "GIF";
           break;
case 9:
     strImgFileType = "WIF";
           break;
}
frmInfo.m_FileType = strImgFileType;
// Read the CompressionType property and translate the value to a
// corresponding string.
switch (ImgAdmin1.GetCompressionType())
{
case 0:
     strImgCompType = "Unknown";
           break;
case 1:
     strImgCompType = "No Compression";
           break;
case 2:
     strImgCompType = "Group3(1D)";
           break;
case 3:
     strImgCompType = "Group3(Modified Huffman)";
           break;
case 4:
     strImgCompType = "PackBits";
           break;
case 5:
     strImgCompType = "Group4(2D)";
           break;
case 6:
     strImgCompType = "JPEG";
           break;
case 7:
     strImgCompType = "RBA";
           break;
case 8:
     strImgCompType = "Group3(2D)";
           break;
case 9:
     strImgCompType = "LZW";
           break;
}
frmInfo.m_CompType = strImgCompType;
// Read the PageType property and translate the value to a
// corresponding string.
switch (ImgAdmin1.GetPageType())
{
case 0:
     strImgPageType = "Unknown";
           break;
case 1:
     strImgPageType = "Black and White";
           break;
```

```
case 2:
     strImgPageType = "4 bit grayscale";
           break;
case 3:
     strImgPageType = "8 bit grayscale";
           break;
case 4:
     strImgPageType = "4 bit palettized";
           break;
case 5:
     strImgPageType = "8 bit palettized";
           break;
case 6:
     strImgPageType = "24 bit RGB";
           break;
case 7:
     strImgPageType = "24 bit BGR";
           break;
}
frmInfo.m_PageType = strImgPageType;
// Determine the dimensions of the image page.
long lInfo = ImgAdmin1.GetImageHeight();
frmInfo.m_Height.Format("%i",lInfo);
lInfo = ImgAdmin1.GetImageWidth();
frmInfo.m_Width.Format("%i",lInfo);
// Determine the X and Y resolution of the image page.
lInfo = ImgAdmin1.GetImageResolutionX();
frmInfo.m_XRes.Format("%i",lInfo);
lInfo = ImgAdmin1.GetImageResolutionY();
frmInfo.m_YRes.Format("%i",lInfo);
// Determine the number of pages in the file.
lInfo = ImgAdmin1.GetPageCount();
frmInfo.m_PageCount.Format("%i",lInfo);
// Show the form with the image attributes.
```

Data Type Long.

Remarks The file Open or Save As dialog box is created by the **ShowFileDialog** method. See page 670 for an example of this dialog box.

If no filters are specified by the **Filter** property and the **FilterIndex** property is set to 0

ForceFileLinking1x Property

Description Returns or sets how file pages being added to a document residing in a 1.x repository are handled.

Depending on the setting of the **ForceFileLinking1x** property, the Admin control's **Append**, **Replace**, and

```
        Case 1
            strImgFileType = "TIF"
        Case 2
            strImgFileType = "AWD"
        Case 3
            strImgFileType = "BMP"
        Case 4
            strImgFileType = "PCX"
        Case 5
            strImgFileType = "DCX"
        Case 6
            strImgFileType = "JPG"
        Case 7
            strImgFileType = "XIF"
        Case 8
            strImgFileType = "GIF"
        Case 9
            strImgFileType = "WIF"
    End Select
    frmInfo.lblImgInfo(0).Caption = strImgFileType
    'Read the CompressionType property and translate the value to a
    'corresponding string.
    Select Case ImgAdmin1.CompressionType
        Case 0
            strImgCompType = "Unknown"
        Case 1
            strImgCompType = "No Compression   strImgFileType = "a2lecion   strg s.a5ee3Z9
```

```
Case 2
    strImgPageType = "4 bit grayscale"
Case 3
    strImgPageType = "8 bit grayscale"
Case 4
    strImgPageType = "4 bit palettiq5
```

```
case 3:
     strImgFileType = "BMP";
           break;
case 4:
     strImgFileType = "PCX";
           break;
case 5:
     strImgFileType = "DCX";
           break;
case 6:
     strImgFileType = "JPG";
           break;
case 7:
     strImgFileType = "XIF";
           break;
case 8:
     strImgFileType = "GIF";
           break;
case 9:
     strImgFileType = "WIF";
           break;
}
frmInfo.m_FdXui0e0´´n"WIF";
           break;
```

```
    Case 6
        strImgFileType = "JPG"
    Case 7
        strImgFileType = "XIF"
    Case 8
        strImgFileType = "GIF"
    Case 9
        strImgFileType = "WIF"
End Select
frmInfo.lblImgInfo(0).Caption = strImgFileType
'Read the CompressionType property and translate the value to a
'corresponding string.
Select Case ImgAdmin1.CompressionType
    Case 0
        strImgCompType = "Unknown"
    Case 1
        strImgCompType = "No Compression"
    Case 2
        strImgCompType = "Group3(1D)"
    Case 3
        strImgCompType = "Group3(Modified Huffman)"
    Case 4
        strImgCompType = "PackBits"
    Case 5
        strImgCompType = "Group4(2D)"
```

```
End Select
    frmInfo.lblImgInfo(6).Caption = strImgPageType
```

```
case 7:
     strImgFileType = "XIF";
```

Chapter 8

```
case 6:
     strImgFileType = "JPG";
           break;
case 7:
     strImgFileType = "XIF";
           break;
case 8:
     strImgFileType = "GIF";
           break;
```

```
        Case 6
            strImgPageType = "24 bit RGB"
        Case 7
            strImgPageType = "24 bit BGR"
End Select
    frmInfo.lblImgInfo(6).Caption = strImgPageType
    'Determine the dimensions of the image page.
    frmInfo.lblImgInfo(2).Caption = ImgAdmin1.ImageHeight
    frmInfo.lblImgInfo(3).Caption = ImgAdmin1.ImageWidth
    'Determine the X and Y resolution of the image page.
    frmInfo.lblImgInfo(4).Caption = ImgAdmin1.ImageResolutionX
    frmInfo.lblImgInfo(5).Caption = ImgAdmin1.ImageResolutionY
    'Determine the number of pages in the file.
    frmInfo.lblImgInfo(7).Caption = ImgAdmin1.PageCount
    'Show the form with the image attributes.
    frmInfo.Show
End Sub
```

PageCount Example – VC++

This example shows how you can use the CompressionType, FileType, ImageHeight, ImageResolutionX, ImageResolutionY, ImageWidth and PageCount properties to request information about a file. Attributes for specific page of a specific file will be displayed.

```
case 6:
     strImgFileType = "JPG";
           break;
case 7:
     strImgFileType = "XIF";
           break;
case 8:
     strImgFileType = "GIF";
           break;
case 9:
     strImgFileType = "WIF";
           break;
}
frmInfo.m_FileType = strImgFileType;
// Read the CompressionType property and translate the value to a
// corresponding string.
switch (ImgAdmin1.GetCompressionType())
{
case 0:
```

PageType Property

Description

```
'Load the form where we will display file attributes.
Load frmInfo
'Read the Filetype property and translate the value to a
'corresponding string.
Select Case ImgAdmin1.FileType
    Case 0
        strImgFileType = "76 TD1˝[(    .re 218   play file attributes.)]TJ1˝0 -1.2727
```

```
case 6:
     strImgCompType = "JPEG";
           break;
case 7:
     strImgCompType = "RBA";
           break;
case 8:
     strImgCompType = "Group3(2D)";
           break;
case 9:
     strImgCompType = "LZW";
           break;
}
frmInfo.m_CompType = strImgCompType;
```

```
ImgAdmin1.CancelError = True
```

```
// Set filename to be printed to the displayed file. If this property
// is not set, the dialog box will not display.
ImgAdmin1.SetImage (ImgEdit1.GetImage());
VARIANT vhWnd; V_VT(&vhWnd) =  VT_I4;
V_I4(&vhWnd) = (long)m_hWnd;
ImgAdmin1.ShowPrintDialog (vhWnd);
// Print the image using the parameters obtained from the print
// dialog box (ex. start page, end page etc.).
VARIANT vStart, vEnd, vOutputFormat, vAnnotations, evt;
```

```
    ImgEdit1.Image = "D:\image2\4page.tif"
    ImgEdit1.Display
    'Reset NumCopies in case user printed multiple copies last time.
    ImgAdmin1.PrintNumCopies = 1
    'If CancelError is true, an error is generated if user presses
    'cancel. Trap the error to avoid trying to print the file.
    ImgAdmin1.CancelError = True
    'Set filename to be printed to the displayed file. If this
    'property is not set the dialog box will not display.
    ImgAdmin1.Image = ImgEdit1.Image
    ImgAdmin1.ShowPrintDialog Form1.hWnd
    'Print the image using the parameters obtained from the print dialog box
    '(e.g., start/end page).
    ImgEdit1.PrintImage ImgAdmin1.PrintStartPage, ImgAdmin1.PrintEndPage,
    ➥    ImgAdmin1.PrintOutputFormat, ImgAdmin1.PrintAnnotations
PrintErr:
    'User pressed the cancel button.
        Exit Sub
End Sub
```

PrintNumCopies Example – VC++

This example uses the ShowPrintDialog method to enable the user to specify printing parameters. The image displayed in the Image Edit control is then printed.

```
void CAdmin1Dlg::OnPrint()
{
    // Display an image.
    ImgEdit1.SetImage ("D:\\image2\\4page.tif");
    ImgEdit1.Display();
    // Reset NumCopies in case user printed multiple copies last tt() 1.S1 8mTJı˝0 i7(
```

Subject Example – VC++

```
        VARIANT vhWnd; V_VT(&vhWnd) =  VT_I4;
        V_I4(&vhWnd) = (long)m_hWnd;
        ImgAdmin1.ShowFileDialog (0,vhWnd);   // OpenDlg
        strAppendFile = ImgAdmin1.GetImage();
        // Restore image property to the original filename because this property
        // must contain the destination file.
        ImgAdmin1.SetImage (strOrigFile);
        // User wants to append 3 pages starting with page 1.
        VARIANT evt;
        evt.vt = VT_ERROR;   // Set to error for optional parameter
        ImgAdmin1.Append (strAppendFile, 1, 3, evt, evt);
    }
```

Browse1x Method

Description

GetSysFileType Method

Description

Returns One of the following file types is returned:

GetUniqueName Method

Ch 8

Chapter 8

Rename Method

Description Renames a file, a document, or a directory in an Eastman Imaging 1.x Server file or document manager namespace.

Available With

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

Arguments The SetSystemFileAttributes method has the following arguments:

Parameter	Data Type	Description
PageType	Integer	Specifies a page type: 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
FileType	Integer	Specifies a file type value: 0 — Unknown 1 — TIF 2 — AWD (Windows 95 and 98 only) 3 — Bitmap (BMP) 4 — PCX 5 — DCX 6 — JPEG 7 — XIF 8 — GIF
CompressionType	Integer	Specifies compression information: 0 — Unknown 1 — No compression 2 — Group 3 1D FAX 3 — Group 3 Modified Huffman 4 — Packbits 5 — Group 4 2D FAX 6 — JPEG 7 — Reserved 8 — Group 3 2D FAX 9 — LZW

```
    ImgAdmin1.ShowFileDialog OpenDlg, Form1.hWnd
    'Determine if a file was selected or cancel was pressed.
    If ImgAdmin1.Image = "" Then Exit Sub
    'Set the image properties in the Image Edit and Thumbnail
    'controls to the name of the file selected in the dialog box.
    ImgEdit1.Image = ImgAdmin1.Image
    ImgThumbnail1.Image = ImgAdmin1.Image
    'Display the image in the ImgEdit and Thumbnail control.
    ImgEdit1.Display
End Sub
```

Example 2 — SaveAs

This example saves the displayed image with a new filename using the Admin Save dialog and the Image Edit control SaveAs method.

```
Private Sub cmdSaveAs_Click()
    ImgAdmin1.ShowFileDialog SaveDlg
    ImgEdit1.SaveAs ImgAdmin1.Image
End Sub
```

Chapter 8

```
PrintErr:
    'User pressed the cancel button.
        Exit Sub
End Sub
```

ShowPrintDialog Example – VC++

This example uses the ShowPrintDialog method to enable the user to specify printing parameters. The image displayed in the Image Edit control is then printed.

Arguments The VerifyImage method has the following argument:

Returns Boolean.

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

9

Image OCR Control

This chapter describes the properties, methods, and events for the Image OCR control. In addition to the methods and events described in this chapter, the following methods and events, which are described in Chapter 7, apply to the Image OCR control:

- AboutBox method
- GetVersion method
- ReadyStateChange event

In This Chapter

CopytoClipBoard Property

Description Specifies that the OCR results are to be placed on the Windows Clipboard.

Available With

Usage *object*.**CopytoClipBoard**[= {*True*|*False*}]

Data Type Boolean.

CopyToClipBoard Example — VB

This example demonstrates how OCR results can be output to the clipboard instead of to an output file of the supported types.

```
Private Sub cmdCopyToClip_Click()
    'If the output type has been set to Text Document, then copy the output
    'to the clipboard. Else save it to a document and launch the associated
    'application.
    If Imgocr1.OutputType = wiAsciiText Then
        Imgocr1.CopyToClipBoard = True
```

Data Type String.

Remarks The source image must be a TIFF file located on a local or redirected drive, or on a 1.x or 3.x server.

Unless you are performing an OcrFromClipBoard function, you must set this value before processing can be performed.

Image Example — VB

This example demonstrates how various images can be specified by the Image property. It

Data Type Long.

Remarks

Quality Property

Description

RetainPictures Property

Description Specifies whether pictures from the image document being processed are retained in the output.

Available With

Usage *object*.**RetainPictures**[={*True*|*False*}]

Data Type

Remarks

Data Type Long.

Remarks If the TrainingFile property is not set, this property is ignored.

```
    If ret1 = CancelDialog Then Exit Sub   'Do not OCR if user pressed
                                           'Cancel.
    ret2 = Imgocr1.ShowOcrOptions          'Display OCR Options dialog.
    If ret2 = CancelDialog Then Exit Sub   'Do not OCR if user pressed
                                           'Cancel.
    Imgocr1.StartOcr
End Sub
```

ShowOcr Method

Description Displays a dialog box (shown here) for OCR settings.

Available With

√ Imaging for Win0 TD6(alo)14u30.04 554nds 669j01magÓ26

Usage *object*.**ShowOcr**

Returns Boolean.

Remarks The dialog box contains the following fields:

If Interactive training (see page 718) is selected, a training dialog box (shown here) is

OcrComplete Event

Description Signals the completion of OCR processing.

Available With

Usage **Sub** *object*_**OCRComplete**(*status*)

Arguments

Input Quality Types

Scroll Property 741
ShowSetupBeforeScan Property 742

FileType Example — VC++

Image Property

Description Returns or sets the name of the object to which you are scanning.

Available With

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

Data Type

MultiPage Example — VB

This example shows how to insert a scanned page before page 1 of an existing image file.

```
Private Sub cmdInsert_Click()
    'Use the ImgAdmin control to select a file for display in the ImgEdit
    'control.
```

Page Example — VC++

This example shows how to insert a scanned page before page 1 of an existing image file.

```
void CNewscanDlg::OnInsertpage()
{
     // This example shows how to insert a scanned page before page 1 of an
     // existing image. Use the ImgAdmin control to select a file for
     // display in the ImgEdit control.
     VARIANT vhWnd; V_VT(&vhWnd) =  VT_I4;
     V_I4(&vhWnd) = (long)m_hWnd;
     ImgAdmin1.ShowFileDialog(0,vhWnd); // OpenDlg  // 0
     // Check to see if cancel was pressed.
     ImgEdit1.SetImage(ImgAdmin1.GetImage());
     ImgEdit1.Display();
     // For insert or append, set the Scan control’s Image property.
     ImgScan1.SetImage(ImgAdmin1.GetImage());
     // Multipage must be set to True in order to create files with more
     // than one page.
     ImgScan1.SetMultiPage(TRUE);
     ImgScan1.SetPageOption(3);  //InsertPages '3
     ImgScan1.SetPage (1);
     ImgScan1.SetFileType(1); // TIFF '1
     // Scan using the Scan Page dialog box.
     VARIANT vModal; V_VT(&vModal) = VT_BOOL;
     V_BOOL(&vModal) = TRUE;
     ImgScan1.ShowScanPage(vModal);
}
```

PageOption Property

ScanTo Property

Description Returns or sets the destination of the image being scanned.

Available With

Usage *object*.**ScanTo**[=*value*]

Data Type Integer (enumerated).

Remarks

See Also

Chapter 10

```
'MultiPage must be true in order to create files with more than one
'page.
ImgScan1.MultiPage = True
'Create 3 page image files.
ImgScan1.PageCount = 3
'Do not show the scanner’s TWAIN UI.
ImgScan1.ShowSetupBeforeScan
```

See Also GetPageTypeCompressionInfo method, GetPageTypeCompressionType method, SetPageTypeCompressionOpts method.

GetCompressionPreference Example — VB

This example demonstrates how to read and retain all current compression settings. In the event that another application makes changes to these settings at runtime, the saved values can be restored.

```
Private Sub cmdGetCompression_Click()
    'Option Base 1
    Dim lngCompInfo(6) As Long
    Dim intCompType(6) As Integer
    Dim intCompPref, intImgType As Integer
    CompPref = ImgScan1.GetCompressionPreference
    'Using intImgType to loop through 6 times (for each page type) and also
    'as the input parameter to the methods.
    For intImgType = 1 To 6
        intCompType(intImgType) =
        ➥    ImgScan1.GetPageTypeCompressionType(intImgType)
        lngCompInfo(intImgType) =
        ➥    ImgScan1.GetPageTypeCompressionInfo(intImgType)
    Next intImgType
End Sub
```

GetCompressionPreference Example — VC++

This example demonstrates how to read and retain all current compression settings. In the event that another application makes changes to these settings at runtime, the saved values can be restored.

```
void CNewscanDlg::OnGetcompression()
{
    // Option Base 1
    long lCompInfo[6];
    long iCompType[6];
    int iCompPref, iImgType;
    iCompPref = ImgScan1.GetCompressionPreference();
    // Using intImgType to loop through 6 times (for each page type) and
    // also as input parameter to the methods.
    for(iImgType = 0; iImgType < 6 ; iImgType++)
        {
        iCompType[iImgType] =
        ➥    ImgScan1.GetPageTypeCompressionType(iImgType+1);
        lCompInfo[iImgType] =
        ➥    ImgScan1.GetPageTypeCompressionInfo(iImgType+1);
        }
}
```

GetPageTypeCompressionInfo Example — VC++

This example demonstrates how to read and retain all current compression settings. In the event that another application makes changes to these settings at runtime, the saved values can be restored.

```
void CNewscanDlg::OnGetcompression()
{
    // Option Base 1
    long lCompInfo[6];
    long iCompType[6];
    int iCompPref, iImgType;
    iCompPref = ImgScan1.GetCompressionPreference();
    // Using intImgType to loop through 6 times (for each page type)
    // and also as the input parameter to the methods.
    for(iImgType = 0; iImgType < 6 ; iImgType++)
        {
        iCompType[iImgType] =
        ➥    ImgScan1.GetPageTypeCompressionType(iImgType+1);
```

Returns

Remarks

```
        Case 2
            If varScanImgType And IT_GRAY8 Then
               ImgScan1.SetScanCapability CAP_SCAN_IMAGE_TYPE, IT_GRAY8
            Else
               MsgBox "Grayscale image type is not supported by your scanner"
            End If
        Case 3
            If varScanImgType And IT_RGB Then
               ImgScan1.SetScanCapability CAP_SCAN_IMAGE_TYPE, IT_RGB
            Else
               MsgBox "Color image type is not supported by your scanner"
            End If
        End Select
    End Sub
```

```
case 2:
    if(V_I4(&vScanImgType) & IT_GRAY8)
            {
                 V_I2(&vCap)= IT_GRAY8;
                 ImgScan1.SetScanCapability (CAP_SCAN_IMAGE_TYPE,vCap);
             }
    else
       AfxMessageBox ("Grayscale image type is not supported by your
       ➥    scanner");
            break;
case 3:
    if(V_I4(&vScanImgType) & IT_RGB)
            {
                      {
```

OpenScanner Example — VC++

Returns Boolean.

Remarks This is done by checking the software (drivers and .dlls), not the hardware.

ScannerAvailable Example — VB

See Also

ReadyState Property 803
ScrollDirection Property 804
SelectedThumbCount Property 805
ThumbBackColor Property 806
ThumbCaption Property 807
ThumbCaptionColor Property 808

AutoSelect Property

Des025ty

BackColor Property

Description Returns or sets the background color of the entire control.

Available With

Usage *object*.**BackColor**[=*color*]

Data Type Long.

Remarks Use the RGB format (see page 540). The default value is light gray — RGB (192, 192, 192).

For this example, the **EnableDragDrop** property has been set to DropFileDropDragLeftRight.

Chapter 11

MousePointer Example — VB

This example shows how to set a custom mouse pointer for the control.

```
Private Sub cmdChangeMouse_Click()
   Dim strMIcon As String
    strMIcon = "C:\Program Files\DevStudio\VB\samples\PGuide\Optimize
    ➥    \Litening.ico"
    ImgThumbnail1.MouseIcon = LoadPicture(strMIcon)
    ImgThumbnail1.MousePointer = wiMPCustom  'Custom pointer = 99
End Sub
```

ReadyState Property

Description

```
Case 0
```

```
        End With
        ImgThumbnail1.ThumbCaptionFont = cptFont
        'Caption color is also set in Font dialog.
        ImgThumbnail1.ThumbCaptionColor = CommonDialog1.Color
        Exit Sub
    Err_Handle:
        G
```

ThumbDropNames Example — VB

This example demonstrates how the Thumbnail control can be updated when images are

```
                ImgThumbnail1.DeleteThumbs i, 1
                iPagesDeleted = iPagesDeleted + 1
            End If
        Next i
        'Display a message regarding the status of the delete attempt.
        If iPagesDeleted <> ThumbsToDelete Then
MsgBox
```

```
Case 1 To 4
    ImgThumbnail1.DisplayThumbs
```

GetManualThumbFilename Method

Description

InsertThumbs Example — VB

This example demonstrates how the Thumbnail control can be updated when images are dropped onto it. A message box is displayed detailing the source file and pages as well as the insertion thumbnail for each thumbnail droppped onto the control. The source can be a drop from Explorer, Imaging for Windows, or selected thumbnails from another Thumbnail control.

ScrollThumbs Method

Description Specifies a direction and amount for scrolling.

Available With

Usage *object*.**ScrollThumbs**(*Direction*,*Amount*)

Arguments

Chapter 11

```
Private Sub cmdControlSize_Click()
    Dim lMinSize, lMaxSize, lVSize, lHSize As Long
    Dim bScroll As Boolean
    Dim iNonScrollThumbCount As Long
    bScroll = True
    iNonScrollThumbCount = 2        'Number of thumbnails desired in
                                    'non-scrolling direction.
    ImgThumbnail1.ScrollDirection = Horizontal
    ImgThumbnail1.UISetThumbSize    'User sets thumb size interactively.
    'Determine min and max control size taking a scrollbar into account.
    lMinSize = ImgThumbnail1.GetMinimumSize(iNonScrollThumbCount, bScroll)
    lMaxSize = ImgThumbnail1.GetMaximumSize(iNonScrollThumbCount, bScroll)
    lVSize = (lMinSize + lMaxSize) / 2  'Average of the min & max
                                        'nonscrolling size
```

Appendix A

Sample Applications

This section describes the Imaging ActiveX sample applications.

The code in each sample application is highly organized, commented, and written using Hungarian notation. There are eight sample applications in the following categories:

prevent the top and bottom of the splitter bar from being visible as you drag it. See the code within the `kdkImgEdit1_DragDrop` and `kdkImgThumbnail1_DragDrop` event procedures.

Imaging Flow Samples

Image Admin Control

Extender Properties, Methods, and Events of the Image Admin Control

	Microsoft Products				Eastman Software Products		
COMPONENT	**Win 95**	**NT 4.0**	**Win 98**	**NT 5.0**	**Pro 1.0**	**Pro 1.1**	**Pro 2.0**
PROPERTIES							
Index	RO	RO	RO	RO	RO	RO	RO
Name	RT RO	RT RO	RT RO	RT RO	RT RO	RT RO	RT RO
Object	RO	RO	RO	RO	RO	RO	RO
Parent	RO	RO	RO	RO	RO	RO	RO
Tag	A	A	A	A	A	A	A
METHODS							
No Methods							
EVENTS							
No Events							

Image Annotation Tool Button Control (continued)

Microsoft

Extender Properties, Methods, and Events of the Image Annotation Tool Button Control

Image Edit Control

This table lists the properties, methods, and events of the Image Edit control by product version.

	Microsoft Products				Eastman Software Products		
COMPONENT	**Win 95**	**NT 4.0**	**Win 98**	**NT 5.0**	**Pro 1.0**	**Pro 1.1**	**Pro 2.0**
PROPERTIES							
AnnotationBackColor	A	A	A	A	A	A	A

Image Edit Control (continued)

Microsoft Products	Eastman Software Products

Image Edit Control (continued)

Image Edit Control (continued)

Appendix C

Image Scan Control (continued)

	Microsoft Products				Eastman Software Products		
COMPONENT	Win 95	NT 4.0	Win 98	NT 5.0	Pro 1.0	Pro 1.1	Pro 2.0
METHODS							

Extender Properties, Methods, and Events of the Image Scan Control

	Microsoft Products	Eastman Software Products

Image Thumbnail Control (continued)

	Microsoft Products				Eastman Software Products		
COMPONENT	**Win 95**	**NT 4.0**	**Win 98**	**NT 5.0**	**Pro 1.0**	**Pro 1.1**	**Pro 2.0**
EVENTS							
Standard Events							
KeyUp	A	A	A	A	A	A	A
Standard Events in							

Extender Properties, Methods and Events of the Image Thumbnail

B

B

Tips and Tricks

This section provides some tips and tricks for using the

Image File Management Tips

Provide file type and page property options to your users

Results of Varying Image File Type and Page Property Options

File Type	Color Type	Compression Applied	Resolution 100 x 100	Resolution 200 x 200	Resolution 300 x 300
AWD	Black & White	Microsoft proprietary	81 KB	120 KB	149 KB
BMP	Black & White	Not available	114 KB	448 KB	1.0 MB
	Pallettized 8-bit	Not available	898 KB	3.5 MB	7.9 MB
	BGR 24-bit	Not available	2.6 MB	10.5 MB	23.6 MB
TIFF	Black & White	None	112 KB	449 KB	1.0 MB
		Group3 (1d)	68 KB	134 KB	226 KB
		Group3 Mod. Huffman	67 KB	132 KB	223 KB
		Group4 (2d)	68 KB	92 KB	115 KB
		PackBits	59 KB	193 KB	389 KB
	Gray Scale 4-bit	None	449 KB	1.8 MB	3.9 MB
		LZW	132 KB	261 KB	579 KB
	Gray Scale 8-bit	None	898 KB	3.5 MB	7.9 MB
		LZW	585 KB	1.5 MB	2.5 MB

[1] By adjusting JPEG resolution and quality compression options, you can increase or decrease the file size by 20 to 30%. In our newsletter example, the JPEG compression option applied was medium resolution and medium quality. The image file would be larger if we applied high resolution and high quality and smaller if we applied low resolution and low quality.

To provide this capability, invoke the

If users respond with Yes

If users want to include graphics in the OCR results, set the **AnnotationOcrType** property to `wiOcrTypePicture` (literal `1`). Then have your users draw OCR picture zones over the graphics they want to include.

To OCR the entire page — Set the

Index

Symbols

_DImgAdmin 239
_DImgAnnTool 239
_DImgEdit 239
_DImgocr 239
_DImgScan 239
_DImgThumbnail 239

A

Access on-line help. *See* on-line help
accessing Imaging 1.x 170
accessing Imaging 3.x 170
action method 257, 262– 265
Active X controls

DragHand 857
f_InitializeApp 45
logging onto server 173, 181, 189
OCR 861
Page with Header 859
querying 1.x server documents 185
querying 3.x server documents

Image Scan 161
Image Thumbnail 161
ImageScaleHeight 120
ImageScaleWidth 120
ImageView 35, 36
Init1xFindDir 185
LastSelectedThumb 161
Left

S