



Disclaimer of Warranties and Limitation of Liabilities

Nothing contained herein modifies or alters in

Contents

Developer's Guide

About This Guide

Purpose xiv

Prerequisites xiv

RelateAbout Th416.02 022 108 553.3197 Tm0.09 0.4 1 scn-0.0001 T

Example 32
The Automation From Excel Project 32

Visible Property	52
WebToolBarVisible Property	53
Width Property	53
Zoom Property	53
Application Object Methods	54
CreateImageViewerObject Method	54
FitTo Method	54

- Image Selection 106
- Annotation Selection 106
- Example 107
- Copy Image Project 107
- Printing An Image 110
 - Image Printing Defined 110
 - Example 110
 - Print Image Project 111
- Scanning an Image Using a Template 117
 - Template Scanning Defined 117
 - Example 118
 - Template Scan Project 120
- Managing an Image File Using Thumbnails 128
 - Thumbnails Defined 128.9(a)-5(n)3.1()-6(Project)-4.3()-6(120)5 apgma

Invoking the Standard Annotation
Tool Palette 213

Purpose

The *Developer's Guide* describes the features of eiStream Imaging for Windows® and provides software developers and MIS professionals with the infor

Imaging Preview

Imaging Preview is a lean version of the Imaging application. It lets users

Image Thumbnail control — The Image Thumbnail control displays and manages thumbnail renditions of individual image pages.

Which to Use: Command-line Interface, OLE, Automation, or

time. Set the appropriate properties or provide end users with drag-and-drop capability so they can select

The code in each sample project is highly organized, heavily commented, and written using Hungarian notation. The sample applications show you how to:

- Create an application that is similar to the standard Imaging application.
- Develop an application that prepares separator pages for scanning several multipage documents in the Imaging Flow application.
- Perform template scanning.
- Use the Image Thumbnail control to create folder-based contact sheets.
- Print a selected portion of an image.
- Use General and Page properties to analyze image files in folders.

What Is Document Imaging?

PageRange Object

The PageRange object represents a range of consecutive pages within an ImageFile object — starting at the

Embedded Server Mode

Imaging for Windows has several Imaging Automation properties and methods to manipulate an embedded image document object.

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

One Page

Then, the

Chapter 3

Remarks The DisplayScaleAlgorithm value can be specified before or after an image is displayed. The

Height Property

Description Sets or returns the distance, in pixels, between the top and bottom edge of the Application object's window. This is a read/write property.

Usage *ApplicationObject*.**Height** [=value]

Data Type Long.

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

Remarks The **ToolBarVisible** property settings are:

Top Property

Description

Remarks The **Visible** property settings are:

WebToolBarVisible Property

Description Sets or returns the visibility of this Application object's web toolbar. This is a read/write

Remarks ZoomOption settings are:

Setting	Description
1	Best fit
2	Fit to width
3	Fit to height
4	Actual size

Help Method

Description Displays the Imaging online Help table of contents.

Usage *ApplicationObject.Help*

Quit Method

Description Closes all open objects and exits the application. The Application object is no longer active or available.

Usage *ApplicationObject.Quit*

ImageFile Object

An ImageFile object represents an image file. An ImageFile object can have

- One Page object, representing the currentl

AppendExistingPages Method

Description Appends specified page(s) to the end of the current ImageFile object. If the page(s) being

Remarks The **CompressionType** property settings are:

Example 'This example returns this page's compression type.
`x = Img.Pages(1).CompressionType`

Height Property

Description Returns this page's height in pixels. This is a read-only property.

Usage *PageObject*.**Height**

Data Type Long.

Example 'This example returns this page's height in pixels.
`x = Img.Pages(1).Height`

ImageResolutionX Property

Description Sets or returns this page's horizontal resolution, in dots-per-inch. An error occurs when a value less than 20 or greater than 1200 dpi is specified. This is a read/write property.

Usage *PageObject*.

Chapter 3

Page Object Methods

The following table lists

Visual Basic

Visual C++

There are two ways to access the Imaging ActiveX Controls on-line help

Demonstration Projects

This section demonstrates how to add a variety of Imaging functions to your applications.

A wide-ranging discussion of every Imaging function is beyond the scope of this

Aesthetics aside, the most important consideration when selecting the color type is file size. The greater the number of data bits per pixel, the

The following compression types are available.

CCITT Group 3 (1d) Fax — Should be used to compress black-and-

Example

Users of your application may want to change the compression type to save disk space.

Scenario

The **ShowPageProperties** method returns an integer that indicates

Example

Printing an Image File

Template Scan Project

■

- Click **Scan Options** to apply compression. The `mnuCompressionOptions_Click()` event in the **Main**

- Click **Select Scanner** to select the scanner you want to use. The `mnuSelectScanner_Click()` event in the

Thumbnail Sorter Project

The Thumbnail Sorter project demonstrates using thumbnails to manage an image file. Specifically, it allows you to:

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

Chapter 4

- Invokes the **DeletePages** method of the Image Admin control, passing to it the:
 - Number of the page to delete (from `lngInsertPage`).
 - 1 (which specifies the number of pages to delete).The

- Invokes the **DeleteThumbs** method of the Image Thumbnail control, passing to it the:
 - Number of the page to delete (from `lngInsertPage`).
 - 1 (which specifies the number of pages to delete).

The **DeleteThumbs** method refreshes the control without the unwanted page.

Chapter 4

Unloading a Multipage Image File

The Unload demonstration project shows how to save the individual pages of a multipage image file as a series of single-page files; in effect, extracting — or *unloading* — the pages of a multipage image file. Before walking through the demonstration proj

Developing Client/Server Applications

The following sections explain each server option setting and related property in detail.

File Location for Document Pages (FileStgLoc1x Property)

When your program saves a server document page, it *may* have to create a

Admin control. Each of these methods creates a new image file with a

Once users select the desired directory or folder, the area below the **Look in** box lists the files or documents contained within it, as illustrated in the following figure.

When users select the desired file or document and then click

Once users select the desired direct

You can also bypass the dialog box and simply pass the user name and password as parameters to the **LoginToServer** method, most likely in response to a user completing and closing a logon dialog box of your own design.

Annotations Defined

The Image Edit and Image Annotation Tool Button controls provide several ways to add annotation functions to your image-enabled applications. Using them you can:

- Create an annotation tool palette of your own design.
- Invoke a single method that displays a fully-functional, **standard annotation tool palette** to your users.
- Implement custom annotations programmatically.

The method you choose depends on the annotation 0.g 48e7(m)ta0 depend1cendl4 b(ance

Users can save annotations separately

Image Edit Control

frm1xCDFD — Enables users to query an Imaging 1.x document volume for documents by location, using the following hierarchy:
Cabinet\Drawer\Folder\Document

frm1xQuery — Enables users to query an Imaging 1.x document volume for documents by name, creation date, modification date, or keyword.

object variable into the **Document Manager** combo box (`cboDocManager`).

The procedure ends the query by setting `objResults` to `Nothing` and by invoking the **ImgQueryEnd** method. Both actions free system resources associated with the query.

The procedure wraps up its work by:

- Showing the **1.x Cabinet\Drawer\Folder\Document** window.
- Displaying the number of document manager databases found.
- Giving focus to the **Document Manager** combo box.

```
Private Sub Form_Load()  
    Dim objResults As Object
```


On the **1.x Cabinet\Drawer\Folder\Document** window, click the desired cabinet in the **Cabinets** combo box. The `cboCabinet_Click()`


```
Private Sub cboDrawer_Click()  
    Dim objResults As Object  
    Dim strSinglePlural As String  
    Dim vntItem As Variant  
    .  
    .  
    .  
    mstrDocManager = cboDocManager.Text  
    mstrCabinet = cboCabinet.Text  
    mstrDrawer = cboDrawer.Text  
    '-----
```

On the **1.x Cabinet\Drawer\Folder\Document**

The procedure wraps up its work by:

- Showing the **1.x Query** window.
- Displaying the number of document manager databases found.
- Giving focus to the

On the **1.x Query** window, click the desired document manager database in the **Document Managers** combo box. The `cboDocManager_Click()`


```

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 = "fi nddocs document " & _
            cboName.List(cboName.ListIndex) & " " & txtName.Text

    ElseIf optQuery(1).Value = True Then  ' Query by Date
        strConvertedDate = ImgAdmin1.ConvertDate(txtDate.Text)
        mstrQuery = "fi nddocs " & _
            cboDateName.List(cboDateName.ListIndex) & " " & _
            cboDate.List(cboDate.ListIndex) & " " & strConvertedDate

    ElseIf optQuery(2).Value = True Then  ' Query by Keyword
        mstrQuery = "fi nddocs keyword " & _
            cboKeyword.List(cboKeyword.ListIndex) & " " & txtKeyword.Text
    End If

    ImgAdmin1.ImgQuery Mid(mstrDocManager, 9), mstrQuery, objResults

    For Each vntItem In objResults
        If vntItem <> "" Then
            lstResults.AddItem vntItem
        End If
    Next
    '-----
    ' End the query.
    '-----

    Set objResults = Nothing
    ImgAdmin1.ImgQueryEnd
    '-----
    ' Display the number of documents found.
    '-----

    If lstResults.ListCount = 0 Then
        lblResults.Caption = SELECT_NONE
    ElseIf lstResults.ListCount = 1 Then
        lblResults.Caption = SELECT_SINGULAR
    Else
        lblResults.Caption = SELECT_PLURAL1 & lstResults.ListCount &
            SELECT_PLURAL2
    End If

End Sub

```

Imaging ActiveX Sample Applications

This appendix describes the Imaging ActiveX sample applications that are available on the media on which your software was distributed.

In This Appendix

Overview	216
----------------	-----

Overview

This section introduces you to the Imaging ActiveX sample applications.

The Imaging ActiveX sample applications are relatively large Visual Basic projects that demonstrate how to use the Imaging ActiveX controls to build comprehensive and useful, image-enabled applications.

It is beyond the scope of this appendix to walk you through each and every application. eiStream WMS, Inc. suggests that you run each one and analyze its code to determine whether you can use it:

- Directly in your applications, or
- As a guide to writing your own, related code.

Requirements

With the exception of the sample application, to compile and run the Imaging ActiveX sample applications, you must use:

- Microsoft Visual Basic 6.0.
- Imaging for Windows.

To compile and run the sample application, you must use:

- Microsoft Visual Basic 6.0 with Service Pack 3 or later.
-

The tracker consists of two functions:

In addition to the general, summary, and page property values, the

images appear in gray scale. Color images continue to appear in color using this setting.

Catch errors properly when working with the ShowFileDialog method

When invoked, the

entire image file, you must apply the **BurnInAnnotations**

