PDF Maker DLL by SkySof Software last updated: 6/9/05

PDF Maker DLL is a powerful ActiveX Dynamic Link Library (DLL) containing many useful PDF related functions. These functions can be used to easily create high-quality, professional Adobe Acrobat PDF files from virtually any file type: Excel worksheets, Word documents, PowerPoint files, Access reports, AutoCAD drawings, image files, text files, etc. You can also create password-protected encrypted PDF files with ease. The encrypted files use high level 128-bit RC4 encryption so you can be assured they are safe from hackers. Also, with PDF Maker DLL you can easily add form fields, comments, and bookmarks to your documents. Our professional, talented staff devoted many hours to create PDF Maker DLL and we think you'll agree that by using PDF Maker DLL in your own application you can save lots of time and money. All functions have been thoroughly tested and optimized to perform as quickly as possible. If you discover any problems or have an idea for a new function please email a detailed description to SkySof Software at kusluski@mail.ic.net

Note: PDF Maker DLL requires Adobe Acrobat Standard or Professional 5 or greater. When installing Acrobat make certain that you also install the Adobe Distiller which is not installed by default.

The cost of PDF Maker DLL is only $99.95 per developer and only $499.95 for a site license. If you want to customize the PDF Maker DLL functions to your liking the complete Visual Basic 6 source code can be purchased for only $999.95. All upgrades are free and PDF Maker DLL may be distributed with your application royalty free.

Click the link below now to purchase PDF Maker DLL:

https://www.regnow.com/softsell/nph-softsell.cgi?item=4459-26

You can save 10% if you purchase PDF Maker DLL through PayPal:

Developer Individual License – only $90.95

https://www.paypal.com/xclick/business=kusluski%40mail.ic.net&item_name=PDF+Maker+DLL+Single+Developer+License&item_number=2&amount=90.95&no_note=1&currency_code=USD

Developer Site License – only $449.95

https://www.paypal.com/xclick/business=kusluski%40mail.ic.net&item_name=PDF+Maker+Developer+Site+License&item_number=6&amount=449.95&no_note=1&currency_code=USD

Complete Visual Basic 6 Source Code – only $900.00

https://www.paypal.com/xclick/business=kusluski%40mail.ic.net&item_name=PDF+Maker+Complete+Visual+Basic+6.0+Source+Code&item_number=5&amount=900.00&no_note=1&currency_code=USD

Important! Before using PDF Maker DLL you should configure the Adobe PDF Printing Preferences:

Note: These instructions pertain specifically to Window's XP. They may vary slightly if you are using a different operating system.

1) Click Window's Start Button on lower left corner of screen
2) Select "Printers and Faxes" from the menu
3) With the mouse cursor over the printer "Adobe PDF" (or "Acrobat Distiller") click the right mouse button to invoke the popup menu
4) Select "Printing Preferences…" from the menu
5) Uncheck "View Adobe PDF results"
6) Uncheck "Do not send fonts to Adobe PDF"
7) Uncheck "Ask to Replace existing PDF file"
8) All other options should be checked
9) Click the OK Button to save changes

To add the PDF Maker DLL reference to your Visual Basic 6 project:

1) Load Visual Basic 6.
2) From the Project Menu select "References..."
3) Select reference "PDFmaker" and click the OK Button.

To add the PDF Maker DLL reference to your VB .NET project:

1) Load Visual Basic .NET. and create a new project
2) From the Project Menu select "Add Reference…"

3) Select the COM tab
4) Scroll down the list to find the PDFMaker component
5) Click on the item and click the Select Button
6) Click the OK Button

To add the PDF Maker DLL reference to your VBA for Excel project:

1) Press ALT+F11 from Excel’s main screen to call the Visual Basic Editor.
2) From the Tools Menu select “References…”
3) Select reference "PDFmaker" and click the OK Button.

To add the PDF Maker DLL reference to your VBA for Word project:

1) Press ALT+F11 from Excel’s main screen to call the Visual Basic Editor.
2) From the Tools Menu select “References…”
3) Select reference "PDFmaker" and click the OK Button.

To add the PDF Maker DLL reference to your VBA for Access project:

1) Press ALT+F11 from Excel’s main screen to call the Visual Basic Editor.
2) From the Tools Menu select “References…”
3) Select reference "PDFmaker" and click the OK Button.

To add the PDF Maker DLL reference to your VBA for Powerpoint project:

1) Press ALT+F11 from Excel’s main screen to call the Visual Basic Editor.
2) From the Tools Menu select “References…”
3) Select reference "PDFmaker" and click the OK Button.

Using PDF Maker DLL in your Visual Basic Application

After adding the PDF Maker DLL reference, place the following code in the General Declarations section of your form:

```
Dim WithEvents oPDFmaker As PDFmaker.CreatePDF
```

The WithEvents keyword exposes events JobDone, JobFail, JobStart, LogMessage, PageNumber, and PercentDone. These events are described in more detail later in this document.

To create an instance of PDF Maker DLL place the following statement in the Form_Load subroutine:

```
Set oPDFmaker = New PDFmaker.CreatePDF
```

Note: All proceeding code in this document is Visual Basic 6 code

The Document Object

The document object represents a valid PDF file. Most of the functions of PDF Maker DLL require a document object as its first parameter. Before using these functions you must first create a document object as such:

```
Dim objDoc As Object
Dim strPDF As String
Dim i As Long
Dim n As Long
Dim a As Variant
Dim s As String
Dim b As Boolean
Screen.MousePointer = vbHourglass
strPDF = App.Path & "\pdfmaker.pdf"
b = oPDFmaker.CloseAcrobat 'ensure Acrobat is released from memory by closing any hidden instances of Acrobat
Set objDoc = oPDFmaker.CreateDocObject(strPDF) ‘create the document object
If objDoc Is Nothing Then
   MsgBox "Unable to create object. You must have Adobe Acrobat Standard or Professional installed."
Else
```

```
   n = oPDFmaker.GetBookmarks(objDoc, a) ‘document object is passed to function
   oPDFmaker.CloseDocObject objDoc ‘close the document object and remove Acrobat from memory
   For i = LBound(a) To UBound(a)
       s = s & a(i) & vbCrLf
   Next
End If
Screen.MousePointer = vbDefault
MsgBox s, vbOKOnly, n & " bookmarks"
```

Creating Acrobat forms

With PDF Maker DLL you can easily create Acrobat forms programmatically. Acrobat forms can then be uploaded to a web server for use. Acrobat forms have many advantages over HTML pages. PDF Maker DLL contains many functions for creating and manipulating the following form field types: text box, check box, button, radio button, combo box, list box, and signature. Before creating an Acrobat form you must create a blank PDF file. The following source code demonstrates how to create a form with calculating text fields:

```
Dim strPDF As String
Dim strDOC As String

On Error Resume Next

Screen.MousePointer = vbHourglass

Call oPDFmaker.CloseAcrobat 'ensure Acrobat is closed

strPDF = App.Path & "\blank.pdf"
strDOC = App.Path & "\blank.doc" ‘this is a Word document consisting of one blank page

oPDFmaker.CreatePDFfromWord strPDF, strDOC

If Err.Number Then
   MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
   MsgBox "Word must be installed on this machine.", vbInformation, "Message"
End If

If Err.Number = 0 Then

   Dim objDoc As Object
   Set objDoc = oPDFmaker.CreateDocObject(strPDF) ‘create document object
   If objDoc Is Nothing Then
      MsgBox "Unable to create object. You must have Adobe Acrobat Standard or Professional installed."
   Else
      Dim a(3) As Variant
      Dim varColor(3) As Variant
      Dim varRadioItems(1) As String

      'set location
      a(0) = 10 'lower left X
      a(1) = 765 'lower left Y
      a(2) = 270 'upper right X
      a(3) = 715 'upper right Y

      'Create Labels - actually read-only Text Fields with transparent border
      varColor(0) = "G"
      varColor(1) = 1
      varColor(2) = 0
      varColor(3) = 0
      oPDFmaker.CreateTextField objDoc, "Label1", 1, a, pdoAlignLeft, pdoFieldBorderSolid, , False, 0, , pdoFieldDisplayVisible, True, False, , pdoFieldLineWidthNone, False, False, True, False, varColor, , , pdoTimesRoman, 24, , "Enter Number"
      a(1) = a(1) - 70
      a(3) = a(3) - 70
      oPDFmaker.CreateTextField objDoc, "Label2", 1, a, pdoAlignLeft, pdoFieldBorderSolid, , False, 0, , pdoFieldDisplayVisible, True, False, , pdoFieldLineWidthNone, False, False, True, False, varColor, , , pdoTimesRoman, 24, , "Enter Number"
```

```
    a(1) = a(1) - 70
    a(3) = a(3) - 70
    oPDFmaker.CreateTextField objDoc, "Label3", 1, a, pdoAlignLeft, pdoFieldBorderSolid, , False, 0, , pdoFieldDisplayVisible, True, False, , pdoFieldLineWidthNone, False, False, True, False, varColor, , , pdoTimesRoman, 24, , "Result"

    'Create Text Fields
    a(0) = 200 'lower left X
    a(1) = 765 'lower left Y
    a(2) = 370 'upper right X
    a(3) = 715 'upper right Y
    oPDFmaker.CreateTextField objDoc, "Field1", 1, a, pdoAlignLeft, pdoFieldBorderBeveled, 10, False, 0, 2, pdoFieldDisplayNoPrint, True, False, , pdoFieldLineWidthMedium, False, False, False, False, , , , pdoTimesRoman, 24, "Enter number", , pdoOnFocus, "var txtField = event.target; txtField.fillColor = color.red; txtField.textColor = color.white;"
    'Set backcolor to tranparent when field loses focus
    oPDFmaker.SetFieldAction objDoc, "Field1", pdoOnBlur, "var txtField = event.target; txtField.fillColor = color.transparent; txtField.textColor = color.black;"
    'Prevent non-numeric characters in field
    oPDFmaker.SetFieldAction objDoc, "Field1", pdoKeystroke, "var re = /^\d$/;if (event.willCommit == false) { if (re.test(event.change) == false) { app.beep(); event.rc = false; } }"
    a(1) = a(1) - 70
    a(3) = a(3) - 70
    oPDFmaker.CreateTextField objDoc, "Field2", 1, a, pdoAlignLeft, pdoFieldBorderBeveled, 10, False, 1, 3, pdoFieldDisplayNoPrint, True, False, , pdoFieldLineWidthMedium, False, False, False, False, , , , pdoTimesRoman, 24, "Enter number", , pdoOnFocus, "var txtField = event.target; txtField.fillColor = color.red; txtField.textColor = color.white;"
    oPDFmaker.SetFieldAction objDoc, "Field2", pdoOnBlur, "var txtField = event.target; txtField.fillColor = color.transparent; txtField.textColor = color.black;"
    oPDFmaker.SetFieldAction objDoc, "Field2", pdoKeystroke, "var re = /^\d$/;if (event.willCommit == false) { if (re.test(event.change) == false) { app.beep(); event.rc = false; } }"
    a(1) = a(1) - 70
    a(3) = a(3) - 70
    oPDFmaker.CreateTextField objDoc, "Field3", 1, a, pdoAlignLeft, pdoFieldBorderBeveled, 10, False, 2, , pdoFieldDisplayVisible, True, False, , pdoFieldLineWidthMedium, False, False, True, False, , , , pdoTimesRoman, 24, "Enter text here", , pdoCalculate, "var f1 = this.getField('Field1');var f2 = this.getField('Field2');var f = this.getField('Radio');if(f.isBoxChecked(0)) event.value = f1.value + f2.value;else event.value = f1.value - f2.value;"

    a(1) = a(1) - 70
    a(3) = a(3) - 70

    'Create Button
    'set color to yellow
    'varColor(0) = "RGB"
    'varColor(1) = 1
    'varColor(2) = 1
    'varColor(3) = 0.8

    oPDFmaker.SetColor pdoYellow, varColor

    oPDFmaker.CreateButtonField objDoc, "Button1", 1, a, "Calculate", pdoFieldBorderBeveled, , , True, pdoFieldDisplayVisible, varColor, pdoFieldLineWidthMedium, False, , , , pdoTimesRoman, 20, "Button Field", pdoMouseDown, "this.calculateNow();"

    a(0) = a(0) + 180
    a(2) = a(2) + 180

    'Create a submit button to submit form data to a web page for processing
    oPDFmaker.CreateButtonField objDoc, "Button2", 1, a, "Submit", pdoFieldBorderBeveled, , , True, pdoFieldDisplayVisible, varColor, pdoFieldLineWidthMedium, False, , , , pdoTimesRoman, 20, "Button Field", pdoMouseDown, "this.submitForm('http://www.website.com/langerhans.asp', false, true);"
   'Create Radio Buttons
    varColor(0) = "G"
    varColor(1) = 1
    a(0) = 370 'lower left X
    a(1) = 755 'lower left Y
    a(2) = 420 'upper right X
    a(3) = 735 'upper right Y
    varRadioItems(0) = "Add"
    varRadioItems(1) = "Subtract"
    oPDFmaker.CreateTextField objDoc, "Label4", 1, a, pdoAlignRight, pdoFieldBorderSolid, , False, 0, , pdoFieldDisplayVisible, True, False, , pdoFieldLineWidthNone, False, False, True, False, varColor, , , , 24, , "+"
    a(0) = a(0) + 40 'lower left X
    a(2) = a(2) + 40 'upper right X
    oPDFmaker.CreateRadioButtonField objDoc, "Radio", 1, a, varRadioItems, pdoFieldBorderBeveled, varRadioItems(0), pdoFieldDisplayVisible, , pdoFieldLineWidthMedium, False, False, , , , , , varRadioItems(0), False
    a(0) = a(0) + 40 'lower left X
    a(2) = a(2) + 40 'upper right X
    oPDFmaker.CreateTextField objDoc, "Label5", 1, a, pdoAlignRight, pdoFieldBorderSolid, , False, 0, , pdoFieldDisplayVisible, True, False, , pdoFieldLineWidthNone, False, False, True, False, varColor, , , , 24, , "-"
    a(0) = a(0) + 40 'lower left X
    a(2) = a(2) + 40 'upper right X
    oPDFmaker.CreateRadioButtonField objDoc, "Radio", 1, a, varRadioItems, pdoFieldBorderBeveled, varRadioItems(1), pdoFieldDisplayVisible, , pdoFieldLineWidthMedium, False, False, lngGlyphStyle:=pdoCircle

    Dim varFields() As String
    Dim t As Integer
    t = oPDFmaker.GetFields(objDoc, varFields)
    Dim i As Integer
    Dim s As String
    For i = 0 To UBound(varFields)
      s = s & varFields(i) & vbCrLf
    Next
    MsgBox s, vbOKOnly, t & " fields"

    'oPDFmaker.SetFieldAction objDoc, "Field1", pdoMouseDown, "app.alert('clicked.');"

    '0 = save changes only
    '1 = save entire file
    objDoc.Save 0, strPDF

    'oPDFmaker.SubmitFormToURL objDoc, "http://www.Langerhans.gov/orders.asp"

    oPDFmaker.CloseDocObject objDoc

    oPDFmaker.OpenPDF strPDF

  End If
End If

Screen.MousePointer = vbDefault
```

JavaScript

The programming language JavaScript can be used to empower Acrobat forms. With JavaScript you can create sophisticated forms with interactive interfaces, implement calculated form fields, and submit form data to a web server. These are only a few things you can do within your Acrobat documents using JavaScript. The extent to which you can use JavaScript in your documents is virtually limitless. There are four types of JavaScript in an Acrobat document:

1) Form Field – Attached to form fields. Use function SetFieldAction to set these JavaScripts.
2) Document – Associated with the opening of a Acrobat document. Use function AddScript to set this JavaScript.
3) Document Action – Executed when a predefined event occurs within an Acrobat document. Use function SetDocumentAction to set these JavaScripts.
4) Page – Associated with a specific page within an Acrobat document. Use function SetPageAction to set these JavaScripts.

For a more complete reference on using JavaScript in your documents consider purchasing the indispensable book “Extending Acrobat Forms With JavaScript” by John Deubert (Adobe Press). John Deubert’s web site is http://www.acumentraining.com The PDF Maker DLL Visual Basic Demo code contains several useful JavaScripts. With PDF Maker DLL and JavaScript you will be able to programmatically create awesome Acrobat forms!

Distributing PDF Maker DLL with your application

PDF Maker DLL may be distributed with your application royalty free. When creating the installation program for your application you should include file MSINET.OCX (Microsoft Internet Transfer Control) if you are using the functions UploadFile() and/or DownloadFile(). The file MSINET.OCX can be found in your Window's System directory which is C:\WINDOWS\SYSTEM32 on most machines. If you use functions such as OpenPDF(), CreateBookmarksInPDF(), CreatePDFfromWebPage(), SetOpenPassword() or any other function which opens a PDF file in your application you should include the file PDFOSEND.EXE in your setup program. This file is located in your Window's System Directory. You should install this file on the client's machine in the same directory as the PDF Maker DLL ActiveX DLL or in their Window's System Directory. The file PDFOSEND.EXE is used by PDF Maker DLL to send messages to Adobe Acrobat windows using Window's API.

How to create password-protected/encrypted PDF files

Note: this feature only available with Adobe Acrobat 6 and greater

Passwords and encryption are set from the Adobe Acrobat Printer setup screen. To access these features:

1) Click the Window's Start Button
2) Select "Printers and Faxes" from the menu
3) Right click on "Adobe PDF" to invoke the popup menu
4) Select "Printing Preferences…" from the menu
5) Find "Adobe PDF Security" on the screen and click the drop-down listbox
6) Select "Reconfirm Security for each job" from the list
7) Click the Apply Button to save your changes
8) Click the Edit.. Button to the right of the list box
9) Modify security settings to your liking and click OK Button when done
10) Click OK Button to exit the screen

Now that Adobe Acrobat Distiller has been setup to create password-protected/encrypted files you're good to go! When using functions that create PDF files, such as CreatePDFfromWord(), make sure that the parameter blnApplySecurity is set to True. For example:

```
oPDFmaker.CreatePDFfromWord "C:\MyDir\file.pdf", "C:\MyDir\file.doc", blnApplySecurity:=True
```

If you create a password-protected file and want to open it with the OpenPDF() function you must use the appropriate password:

```
oPDFmaker.OpenPDF "C:\MyDir\file.pdf", strOpenPassword:="opensezme"
```

Events

JobDone – Executes when the Distiller print job has completed.
JobFail – Executes when the Distiller print job fails.
JobStart – Executes when the Distiller print job starts.
LogMessage – Executes while the Distiller print job is processing. Provides detailed information of job.
PageNumber – Executes while the Distiller print job is processing. Provides current page number.
PercentDone – Executes while the Distiller print job is processing. Provides percent of job completed.

Properties

TempFilePath – Read-write. The path where temporary files will be created. Default is C:\Temp

Functions

1. PDF Creation Methods

1.1 CreatePDF - Method to create an Adobe Acrobat (PDF) file from a variety of file types.

Note: This function requires Acrobat 6 and greater.

Arguments:

strSourceFile As String	Source file to use
strPDFFile As String	PDF file to create
Optional intSecsToWaitForPDF As Integer	Seconds to wait for PDF to complete

Example:

```
Dim strFile As String
Dim strPDF As String
Dim i As Long

On Error Resume Next

strFile = App.Path & "\debug.doc"
strPDF = App.Path & "\new.pdf"

'Use this function to quickly create a PDF from a variety of file types including
'DOC, DWG, XLS, GIF, JPG, TIF, etc.

oPDFmaker.CreatePDF strFile, strPDF
If Err.Number = 0 Then
   'Open the PDF
   oPDFmaker.OpenPDF strPDF
End If
```

1.2 CreatePDFfromAccessReport - Method to create an Adobe Acrobat (PDF) file from an Access report.

Requirements: Access and file PDFOSEND.EXE (see page 6 of this document)

Arguments:

strPDFFile As String	PDF file to create
strDatabase As String	Access database to use
strReportName As String	Report to use
Optional strPassword As String	Access database password
Optional strFilter As String	Report filter
Optional strWhereCondition As String	Report where condition
Optional strPDFWriter As String	PDF Writer name. Default: “Acrobat PDFWriter”
Optional blnCloseAcrobat As Boolean	Close Acrobat after creating PDF?

Example:

```
Dim strPDF As String
Dim strMDB As String
Dim strRpt As String
Dim i As Integer

On Error Resume Next
Screen.MousePointer = vbHourglass

strPDF = App.Path & "\invoice.pdf"
'Note: You may need to modify this file path
strMDB = "C:\Program Files\Microsoft Visual Studio\VB98\NWIND.MDB"
strRpt = "invoice"

oPDFmaker.CreatePDFfromAccessReport strPDF, strMDB, strRpt

Screen.MousePointer = vbDefault

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
  'open PDF file
End If
```

1.3 CreatePDFfromAutoCAD - Method to create an Adobe Acrobat (PDF) file from AutoCAD DWG and DXF files.

Requirements: AutoCAD and file PDFOSEND.EXE (see page 6 of this document)

Arguments:

strPDFFile As String	PDF file to create
strDrawing As String	AutoCAD DWG/DXF file to use
Optional varLayout As Variant	Layout to use (can be a name or number greater than 0)
Optional strPDFWriter As String	PDF Writer name. Default: “Acrobat PDFWriter”
Optional blnCloseAcrobat As Boolean	Close Acrobat after creating PDF?

Example:

```
Dim strPDF As String
Dim strDWG As String

Screen.MousePointer = vbHourglass
strPDF = App.Path & "\test.pdf"
strDWG = App.Path & “\test.dwg”
oPDFmaker.CreatePDFfromAutoCAD strPDF, stDWG, “layout1”
'open PDF file
oPDFMaker.OpenPDF strPDF
Screen.MousePointer = vbDefault
```

1.4 CreatePDFfromCrystalReport - Method to create an Adobe Acrobat (PDF) file from a Crystal Report File.

Requirements: Crystal Reports and file PDFOSEND.EXE (see page 6 of this document)

Arguments:

strPDFFile As String	PDF file to create
strCrystalReport As String	Crystal Report File to use
Optional strDistiller As String	Name of Adobe Acrobat Distiller
Optional blnCloseAcrobat As Boolean	Close Acrobat after creating PDF?

Example:

```
Dim strPDF As String
Dim strRPT As String

Screen.MousePointer = vbHourglass
strPDF = App.Path & "\test.pdf"
strRPT = App.Path & “\test.rpt”
oPDFmaker.CreatePDFfromCrystalReport strPDF, strRPT
'open PDF file
oPDFmaker.OpenPDF strPDF
Screen.MousePointer = vbDefault
```

1.5 CreatePDFfromExcel - Method to create an Adobe Acrobat (PDF) file from an Excel worksheet.

Requirements: Excel

Arguments:

StrPDFFile As String	PDF file to create
StrExcelWorkbook As String	Excel workbook file to use
VarExcelWorksheet As Variant	Excel worksheet to use. Can be a name or number
strRange As String	Excel worksheet range to use
Optional strPassword As String	Excel workbook password
Optional strWriteResPassword As String	Excel workbook write reserve password
Optional blnUseGrid As Boolean	Use grid lines. Values: True,False
Optional blnLandscape As Boolean	Page orientation. Values: True,False
Optional strHeader As String	Page header
Optional lngHeaderAlignment As Long	Page header alignment. Values: 0=left, 1=center, 2=right
Optional strFooter As String	Page footer
Optional lngFooterAlignment As Long	Page footer alignment. Values: 0=left, 1=center, 2=right

Optional dblTopMargin As Double	Page top margin in inches
Optional dblLeftMargin As Double	Page left margin in inches
Optional dblRightMargin As Double	Page right margin in inches
Optional dblBottomMargin As Double	Page bottom margin in inches
Optional dblHeaderMargin As Double	Page header margin in inches
Optional dblFooterMargin As Double	Page footer margin in inches
Optional lngFirstPageNumber As Long	Beginning page number
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional blnShowWindow As Boolean	Show job process window. Values: True,False
Optional blnSpoolJobs As Boolean	Spool print jobs. Values: True,False
Optional strJobOptionsFile As String	Print job options file.
Optional blnApplySecurity As Boolean	Apply Distiller security settings. Values: True,False

Example:

```
Dim strPDF As String
Dim strWB As String
Dim strRange As String

On Error Resume Next

Screen.MousePointer = vbHourglass

strPDF = App.Path & "\sample.pdf"
strWB = App.Path & "\sample.xls"
strRange = "A1:E276"

oPDFmaker.CreatePDFfromExcel strPDF, strWB, 1, strRange, , , , , "Page &P", "right", "&D &T", "center", 1.0, 0.5, 0.0, 1.0

Screen.MousePointer = vbDefault

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
  'open PDF file
End If
```

1.6 CreatePDFfromExcel2 - Method to create an Adobe Acrobat (PDF) file from an Excel worksheet.

Requirements: Excel and file PDFOSEND.EXE (see page 6 of this document)

Note: No postscript file created. Creates PDF faster than CreatePDFfromExcel()

Arguments:

StrPDFFile As String	PDF file to create
StrExcelWorkbook As String	Excel workbook file to use
VarExcelWorksheet As Variant	Excel worksheet to use. Can be a name or number
strRange As String	Excel worksheet range to use
Optional strPassword As String	Excel workbook password
Optional strWriteResPassword As String	Excel workbook write reserve password
Optional blnUseGrid As Boolean	Use grid lines. Values: True,False
Optional blnLandscape As Boolean	Page orientation. Values: True,False
Optional strHeader As String	Page header
Optional lngHeaderAlignment As Long	Page header alignment. Values: 0=left, 1=center, 2=right
Optional strFooter As String	Page footer
Optional lngFooterAlignment As Long	Page footer alignment. Values: 0=left, 1=center, 2=right
Optional dblTopMargin As Double	Page top margin in inches
Optional dblLeftMargin As Double	Page left margin in inches
Optional dblRightMargin As Double	Page right margin in inches
Optional dblBottomMargin As Double	Page bottom margin in inches
Optional dblHeaderMargin As Double	Page header margin in inches
Optional dblFooterMargin As Double	Page footer margin in inches

Optional lngFirstPageNumber As Long	Beginning page number
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional blnCloseAcrobat As Boolean	Close Acrobat?

Example:

```
Dim strPDF As String
Dim strWB As String
Dim strRange As String

On Error Resume Next

Screen.MousePointer = vbHourglass

strPDF = App.Path & "\sample.pdf"
strWB = App.Path & "\sample.xls"
strRange = "A1:E276"

oPDFmaker.CreatePDFfromExcel2 strPDF, strWB, 1, strRange, , , , , "Page &P", "right", "&D &T", "center", 1.0, 0.5, 0.0, 1.0

Screen.MousePointer = vbDefault

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
  'open PDF file
End If
```

1.7 CreatePDFfromImageFile - Method to create an Adobe Acrobat (PDF) file from an image file (JPG,GIF,BMP,TIF,PNG,PCX).

Arguments:

strPDFFile As String	PDF file to create
strImageFile As String	Image file to use
Optional strTitle As String	Title to save with PDF file
Optional strAuthor As String	Author to save with PDF file
Optional strSubject As String	Subject to save with PDF file
Optional strKeywords As String	Keywords to save with PDF file

Example:

```
Dim strPDF As String
Dim strJPG As String

On Error Resume Next

Screen.MousePointer = vbHourglass

strPDF = App.Path & "\image.pdf"
'Pratically any type of image file can be used including JPG,TIF,PNG,PCX,GIF,BMP
strJPG = App.Path & "\image1.jpg"

oPDFmaker.CreatePDFfromImageFile strPDF, strJPG, "My PDF file", "Frank Kusluski", "Cute girl", "keywords go here"

Screen.MousePointer = vbDefault

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
  'open PDF file
  oPDFmaker.OpenPDF strPDF
End If
```

1.8 CreatePDFfromMultiAccessReport - Method to create an Adobe Acrobat (PDF) file from multiple Access reports.

Requirements: Access and file PDFOSEND.EXE (see page 6 of this document)

Note: Reports must exist in the same database

Arguments:

strPDFFile As String	PDF file to create
strDatabase As String	Database to use
varReportNames As Variant	array of report names
Optional strPassword As String	Password to use
Optional ByVal strPDFWriter As String	PDF Writer name. Default: “Acrobat PDFWriter”
Optional blnCloseAcrobat As Boolean.	Close Acrobat after creating PDF?

Example:

```
Dim arr(1) As String
arr(0) = “Report1”
arr(1) = “Report2”
oPDFMaker.CreatePDFfromMultiAccess “c:\file.pdf”, “c:\db.mdb”, arr
```

1.9 CreatePDFfromMultiAutoCAD - Method to create an Adobe Acrobat (PDF) file from multiple AutoCAD layouts.

Requirements: AutoCAD and file PDFOSEND.EXE (see page 6 of this document)

Note: Layouts must exist in the same drawing file.

Arguments:

strPDFFile As String	PDF file to create
strDWGFile As String	Drawing file to use
varLayouts As Variant	array of layout names/numbers
Optional ByVal strPDFWriter As String	PDF Writer name. Default: “Acrobat PDFWriter”
Optional blnCloseAcrobat As Boolean	Close Acrobat after creating PDF?

Example:

```
Dim arr(1) As String
arr(0) = “Layout1”
arr(1) = “Layout2”
oPDFMaker.CreatePDFfromMultiAutoCAD “c:\file.pdf”, “c:\drawing.dwg”, arr
```

1.10 CreatePDFfromMultiExcel - Method to create an Adobe Acrobat (PDF) file from multiple Excel worksheets.

Requirements: Excel

Note: Worksheets must exist in the same workbook.

Arguments:

strPDFFile As String	PDF file to create
strExcelWorkbook As String	Excel workbook file to use
varExcelWorksheets As Variant	Excel worksheets to use (array). Can be a name or number
varRanges As Variant	Excel worksheet ranges to use (array)
Optional strPassword As String	Excel workbook password
Optional strWriteResPassword As String	Excel workbook write reserve password
Optional blnUseGrid As Boolean	Use grid lines. Values: True,False
Optional varLandscape As Variant	Page orientation (array). Values: True,False
Optional strHeader As String	Page header
Optional lngHeaderAlignment As Long	Page header alignment. Values: 0=left, 1=center, 2=right
Optional strFooter As String	Page footer
Optional lngFooterAlignment As Long	Page footer alignment. Values: 0=left, 1=center, 2=right

Optional dblTopMargin As Double	Page top margin in inches
Optional dblLeftMargin As Double	Page left margin in inches
Optional dblRightMargin As Double	Page right margin in inches
Optional dblBottomMargin As Double	Page bottom margin in inches
Optional dblHeaderMargin As Double	Page header margin in inches
Optional dblFooterMargin As Double	Page footer margin in inches
Optional varFirstPageNumber As Variant	Beginning page numbers (array)
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional blnShowWindow As Boolean	Show job process window. Values: True,False
Optional blnSpoolJobs As Boolean	Spool print jobs. Values: True,False
Optional blnApplySecurity As Boolean	Apply Distiller security settings. Values: True,False

Example:

```
Dim strPDF As String
Dim strWB As String
Dim varWorksheets(2) As Integer
Dim varRanges(2) As String
Dim varLandscape(2) As Boolean
Dim varFirstPageNumber(2) As Long

On Error Resume Next

Screen.MousePointer = vbHourglass

strPDF = App.Path & "\sample.pdf"
strWB = App.Path & "\sample.xls"
varWorksheets(0) = 1
varWorksheets(1) = 2
varWorksheets(2) = 3
varRanges(0) = "A1:E111"
varRanges(1) = "A1:E131"
varRanges(2) = "A1:E46"
varLandscape(0) = True
varLandscape(1) = False
varLandscape(2) = True
varFirstPageNumber(0) = 1
varFirstPageNumber(1) = 4
varFirstPageNumber(2) = 8

oPDFmaker.CreatePDFfromMultiExcel strPDF, strWB, varWorksheets, varRanges, , , , varLandscape, "Page &P", "right", "&D &T",
"center", 1.0, 0.5, 0.0, 1.0 , , , varFirstPageNumber

Screen.MousePointer = vbDefault

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
  'open PDF file
End If
```

1.11 CreatePDFfromMultiExcel2 - Method to create an Adobe Acrobat (PDF) file from multiple Excel worksheets.

Requirements: Excel and file PDFOSEND.EXE (see page 6 of this document)

Note: No postscript files created. Creates PDF faster than CreatePDFfromMultiExcel()

Arguments:

strPDFFile As String	PDF file to create
strExcelWorkbook As String	Excel workbook file to use
varExcelWorksheets As Variant	Excel worksheets to use (array). Can be a name or number
varRanges As Variant	Excel worksheet ranges to use (array)

Optional strPassword As String	Excel workbook password
Optional strWriteResPassword As String	Excel workbook write reserve password
Optional blnUseGrid As Boolean	Use grid lines. Values: True,False
Optional varLandscape As Variant	Page orientation (array). Values: True,False
Optional strHeader As String	Page header
Optional lngHeaderAlignment As Long	Page header alignment. Values: 0=left, 1=center, 2=right
Optional strFooter As String	Page footer
Optional lngFooterAlignment As Long	Page footer alignment. Values: 0=left, 1=center, 2=right
Optional dblTopMargin As Double	Page top margin in inches
Optional dblLeftMargin As Double	Page left margin in inches
Optional dblRightMargin As Double	Page right margin in inches
Optional dblBottomMargin As Double	Page bottom margin in inches
Optional dblHeaderMargin As Double	Page header margin in inches
Optional dblFooterMargin As Double	Page footer margin in inches
Optional varFirstPageNumber As Variant	Beginning page numbers (array)
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional blnCloseAcrobat As Boolean	Close Acrobat after creating PDF?

Example:

```
Dim strPDF As String
Dim strWB As String
Dim varWorksheets(2) As Integer
Dim varRanges(2) As String
Dim varLandscape(2) As Boolean
Dim varFirstPageNumber(2) As Long

On Error Resume Next

Screen.MousePointer = vbHourglass

strPDF = App.Path & "\sample.pdf"
strWB = App.Path & "\sample.xls"
varWorksheets(0) = 1
varWorksheets(1) = 2
varWorksheets(2) = 3
varRanges(0) = "A1:E111"
varRanges(1) = "A1:E131"
varRanges(2) = "A1:E46"
varLandscape(0) = True
varLandscape(1) = False
varLandscape(2) = True
varFirstPageNumber(0) = 1
varFirstPageNumber(1) = 4
varFirstPageNumber(2) = 8

oPDFmaker.CreatePDFfromMultiExcel2 strPDF, strWB, varWorksheets, varRanges, , , , varLandscape, "Page &P", "right", "&D &T",
"center", 1.0, 0.5, 0.0, 1.0 , , , varFirstPageNumber

Screen.MousePointer = vbDefault

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
  'open PDF file
End If
```

1.12 CreatePDFfromMultiImage - Method to create an Adobe Acrobat (PDF) file from multiple image files (JPG,GIF,BMP,TIF,PNG,PCX).

Arguments:

strPDFFile As String	PDF file to create

varImageFiles As Variant	Image files to use (array)

Example:

```
Dim strPDF As String
Dim img(1) As String

On Error Resume Next

Screen.MousePointer = vbHourglass

strPDF = App.Path & "\multiple.pdf"

img(0) = App.Path & "\image1.jpg"
img(1) = App.Path & "\image2.jpg"

'Create the PDF!
oPDFmaker.CreatePDFfromMultiImage strPDF, img

Screen.MousePointer = vbDefault

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
  'open PDF file
  oPDFmaker.OpenPDF strPDF
End If
```

1.13 CreatePDFfromMultiPDF - Method to create an Adobe Acrobat (PDF) file from multiple PDF files.

Arguments:

strPDFFile As String	PDF file to create
varPDFFiles As Variant	PDF files to use (array)

Example:

```
Dim strPDF As String
Dim strPDF1 As String
Dim strPDF2 As String
Dim strPDF3 As String

Dim pdf(2) As String

On Error Resume Next

Screen.MousePointer = vbHourglass

strPDF = App.Path & "\multiple.pdf"

'Create PDF1
oPDFmaker.CreatePDFfromImageFile App.Path & "\image.pdf", App.Path & "\image1.jpg", "My PDF file", "Frank Kusluski", "Cute girl", "keywords go here"
strPDF1 = App.Path & "\image.pdf"

'Create PDF2
oPDFmaker.CreatePDFfromExcel App.Path & "\sample.pdf", App.Path & "\sample.xls", 1, "October", , , , , "Page &P", "right", "&D &T", "center", 1#, 0.5, 0#, 1#
strPDF2 = App.Path & "\sample.pdf"

'Create PDF3
oPDFmaker.CreatePDFfromWord App.Path & "\debug.pdf", App.Path & "\debug.doc", , , "1, 3, 6-8, 10"
strPDF3 = App.Path & "\debug.pdf"
```

```
'store PDF file names in one dimensional array - order is important!
pdf(0) = strPDF1
pdf(1) = strPDF2
pdf(2) = strPDF3

'finally combine all the PDF files into one PDF file!
oPDFmaker.CreatePDFfromMultiPDF strPDF, pdf

Screen.MousePointer = vbDefault

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
  'open PDF file
  oPDFmaker.OpenPDF strPDF
End If
```

1.14 CreatePDFfromMultiPDFinDir - Method to create an Adobe Acrobat (PDF) file from multiple PDF files in a specific directory.

Arguments:

strPDFFile As String	PDF file to create
strDirectory As String	Directory where PDF files exist
Optional strFiles As String = "*.pdf"	PDF files to use (can use wildcard characters)
Optional lngSortType As Long	Sort field (0=None,1=File Name,2=File Size,3=File Date)
Optional lngSortOrder As Long	Sort order (0=Ascending, 1=Descending)

Example:

```
Dim strPDF As String

On Error Resume Next

Screen.MousePointer = vbHourglass

strPDF = App.Path & "\all.pdf"

'Create the PDF!
oPDFmaker.CreatePDFfromMultiPDFinDir strPDF, App.Path, “f*.pdf”, 1, 0

Screen.MousePointer = vbDefault

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
  'open PDF file
  oPDFmaker.OpenPDF strPDF
End If
```

1.15 CreatePDFfromMultiPS - Method to create an Adobe Acrobat (PDF) file from multiple PostScript (PS) files.

Arguments:

strPDFFile As String	PDF file to create
varPSFiles As Variant	Postscript files to use (stored in one dimensional array)
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional blnShowWindow As Boolean	Show job process window. Values: True,False
Optional blnSpoolJobs As Boolean	Spool print jobs. Values: True,False
Optional strJobOptionsFile As String	Print job options file.
Optional blnApplySecurity As Boolean	Apply Distiller security settings. Values: True,False

Example:

```
Dim strPDF As String
Dim strPS1 As String
Dim strPS2 As String
Dim strPS3 As String
Dim strPS4 As String
Dim strPpt As String
Dim strDoc As String
Dim strTxt As String
Dim strPagesToPrint As String
Dim strWB As String
Dim strRange As String

Dim ps(3) As String

On Error Resume Next

Screen.MousePointer = vbHourglass

strPDF = App.Path & "\multiple.pdf"

strPS1 = App.Path & "\file.ps"
strPpt = App.Path & "\file.ppt"

'create PostScript file from a PowerPoint file
oPDFmaker.CreatePSfromPowerPoint strPS1, strPpt

strPS2 = App.Path & "\sample.ps"
strWB = App.Path & "\sample.xls"
strRange = "A1:E276"

'create PostScript file from an Excel workbook file worksheet
oPDFmaker.CreatePSfromExcel strPS2, strWB, 1, strRange, , , , , "Page &P", "right", "&D &T", "center", 1.0, 0.5, 0.0, 1.0

strPS3 = App.Path & "\debug.ps"
strDoc = App.Path & "\debug.doc"
strPagesToPrint = "1, 3, 6-8, 10"

'create PostScript file from a Word document file
oPDFmaker.CreatePSfromWord strPS3, strDoc, , , strPagesToPrint

strPS4 = App.Path & "\file2.ps" 'note file name of 'file2.ps' - we already used 'file.ps' above
strTxt = App.Path & "\file.txt"

'create PostScript file from a text file
oPDFmaker.CreatePSfromTextFile strPS4, strTxt, False, "arial", False, 8, 255, False, False, False, True, "right", 20.5, 20.5, , 20.5

'store PostScript file names in one dimensional array - order is important!
ps(0) = strPS1
ps(1) = strPS2
ps(2) = strPS3
ps(3) = strPS4

'finally combine all the PostScript files into one PDF file!
oPDFmaker.CreatePDFfromMultiPS strPDF, ps

Screen.MousePointer = vbDefault

If Err.Number Then
   MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
   'open PDF file
End If
```

1.16 CreatePDFfromPowerPoint - Method to create an Adobe Acrobat (PDF) file from a PowerPoint file.

Requirements: Power Point

Arguments:

strPDFFile As String	PDF file to create
strPPDocument As String	PowerPoint file to use
Optional lngStartSlide As Long = -1	PowerPoint start slide to use
Optional lngEndSlide As Long = -1	PowerPoint end slide to use
Optional blnHorizontalOrientation As Boolean	PowerPoint slide orientation. Values: True,False
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional blnShowWindow As Boolean	Show job process window. Values: True,False
Optional blnSpoolJobs As Boolean	Spool print jobs. Values: True,False
Optional strJobOptionsFile As String	Print job options file.
Optional blnApplySecurity As Boolean	Apply Distiller security settings. Values: True,False

Example:

```
Dim strPDF As String
Dim strDoc As String

On Error Resume Next

Screen.MousePointer = vbHourglass

strPDF = App.Path & "\file.pdf"
strDoc = App.Path & "\file.ppt"

oPDFmaker.CreatePDFfromPowerPoint strPDF, strDoc

Screen.MousePointer = vbDefault

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
  'open PDF file
End If
```

1.17 CreatePDFfromPowerPoint2 - Method to create an Adobe Acrobat (PDF) file from a PowerPoint file.

Requirements: Power Point and file PDFOSEND.EXE (see page 6 of this document)

Note: No postscript files created. Creates PDF faster than CreatePDFfromPowerPoint()

Arguments:

strPDFFile As String	PDF file to create
strPPDocument As String	PowerPoint file to use
Optional lngStartSlide As Long = -1	PowerPoint start slide to use
Optional lngEndSlide As Long = -1	PowerPoint end slide to use
Optional blnHorizontalOrientation As Boolean	PowerPoint slide orientation. Values: True,False
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional blnCloseAcrobat As Boolean	Close Acrobat after creating PDF?

Example:

```
Dim strPDF As String
Dim strDoc As String
On Error Resume Next

Screen.MousePointer = vbHourglass
```

```
strPDF = App.Path & "\file.pdf"
strDoc = App.Path & "\file.ppt"

oPDFmaker.CreatePDFfromPowerPoint2 strPDF, strDoc

Screen.MousePointer = vbDefault

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
  'open PDF file
End If
```

1.18 CreatePDFfromPS - Method to create an Adobe Acrobat (PDF) file from a PostScript (PS) file.

Arguments:

strPDFFile As String	PDF file to create
strPSFile As String	Postscript file to use
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional blnShowWindow As Boolean	Show job process window. Values: True,False
Optional blnSpoolJobs As Boolean	Spool print jobs. Values: True,False
Optional strJobOptionsFile As String	Print job options file.
Optional blnApplySecurity As Boolean	Apply Distiller security settings. Values: True,False

Example:

```
Dim strPS As String
Dim strPDF As String
Dim strWB As String
Dim strRange As String

On Error Resume Next

Screen.MousePointer = vbHourglass

strPS = App.Path & "\sample.ps"
strPDF = App.Path & "\sample.pdf"
strWB = App.Path & "\sample.xls"
strRange = "A1:E276"

'create a PostScript file from Excel workbook worksheet range
oPDFmaker.CreatePSfromExcel strPS, strWB, 1, strRange, , , , , "Page &P", "right", "&D &T", "center", 1#, 0.5, 0#, 1#

'create a PDF file from a PostScript file
oPDFmaker.CreatePDFfromPS strPDF, strPS

Screen.MousePointer = vbDefault

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
  'open PDF file
End If
```

1.19 CreatePDFfromTextFile - Method to create an Adobe Acrobat (PDF) file from a text file.

Requirements: Word

Arguments:

strPDFFile As String	PDF file to create
strTextFile As String	Text file to use

Optional blnLandscape As Boolean	Page orientation. Values: True,False
Optional strFont As String	Font name. Default is “Courier”
Optional blnFontBold As Boolean	Font bold. Values: True,False
Optional intFontSize As Integer = 8	Font size
Optional lngFontColor As Long	Font color
Optional blnFontItalic As Boolean	Font italic. Values: True,False
Optional blnFontUnderline As Boolean	Font underline. Values: True,False
Optional blnPageNumberInFooter As Boolean	Place page number in footer. Values: True,False
Optional blnPageNumberInHeader As Boolean	Place page number in header. Values: True,False
Optional lngPageNumberAlignment As Long	Page number alignment. Values: 0=left, 1=center, 2=right
Optional sngTopMargin As Single	Page top margin
Optional sngLeftMargin As Single	Page left margin
Optional sngRightMargin As Single	Page right margin
Optional sngBottomMargin As Single	Page bottom margin
Optional blnFirstPage As Boolean = True	Add page number to first page? Values: True,False
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional blnShowWindow As Boolean	Show job process window. Values: True,False
Optional blnSpoolJobs As Boolean	Spool print jobs. Values: True,False
Optional strJobOptionsFile As String	Print job options file.
Optional blnApplySecurity As Boolean	Apply Distiller security settings. Values: True,False

Example:

```
Dim strPDF As String
Dim strTxt As String

On Error Resume Next

Screen.MousePointer = vbHourglass

strPDF = App.Path & "\file.pdf"
strTxt = App.Path & "\file.txt"

oPDFmaker.CreatePDFfromTextFile strPDF, strTxt, False, "arial", False, 8, 255, False, False, False, True, "right", 20.5, 20.5, , 20.5

Screen.MousePointer = vbDefault

If Err.Number Then
   MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
   'open PDF file
End If
```

1.20 CreatePDFfromTextFile2 - Method to create an Adobe Acrobat (PDF) file from a text file.

Requirements: Word and file PDFOSEND.EXE (see page 6 of this document)

Note: No postscript file created. Creates PDF faster than CreatePDFfromTextFile()

Arguments:

strPDFFile As String	PDF file to create
strTextFile As String	Text file to use
Optional blnLandscape As Boolean	Page orientation. Values: True,False
Optional strFont As String	Font name. Default is “Courier”
Optional blnFontBold As Boolean	Font bold. Values: True,False
Optional intFontSize As Integer = 8	Font size
Optional lngFontColor As Long	Font color
Optional blnFontItalic As Boolean	Font italic. Values: True,False
Optional blnFontUnderline As Boolean	Font underline. Values: True,False
Optional blnPageNumberInFooter As Boolean	Place page number in footer. Values: True,False
Optional blnPageNumberInHeader As Boolean	Place page number in header. Values: True,False
Optional lngPageNumberAlignment As Long	Page number alignment. Values: 0=left, 1=center, 2=right

Optional sngTopMargin As Single	Page top margin
Optional sngLeftMargin As Single	Page left margin
Optional sngRightMargin As Single	Page right margin
Optional sngBottomMargin As Single	Page bottom margin
Optional blnFirstPage As Boolean = True	Add page number to first page? Values: True,False
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional blnCloseAcrobat As Boolean	Close Acrobat after creating PDF?

Example:

```
Dim strPDF As String
Dim strTxt As String

On Error Resume Next

Screen.MousePointer = vbHourglass

strPDF = App.Path & "\file.pdf"
strTxt = App.Path & "\file.txt"

oPDFmaker.CreatePDFfromTextFile2 strPDF, strTxt, False, "arial", False, 8, 255, False, False, False, True, "right", 20.5, 20.5, , 20.5

Screen.MousePointer = vbDefault

If Err.Number Then
   MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
   'open PDF file
End If
```

1.21 CreatePDFfromTextFile3 - Method to create an Adobe Acrobat (PDF) file from a text file.

Requirements: Notepad and file PDFOSEND.EXE (see page 6 of this document)

No postscript file created. Very fast. Creates PDF in minimal time.

Arguments:

strPDFFile As String	PDF file to create
strTextfile As String	Text file to use
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional blnCloseAcrobat As Boolean	Close Acrobat after creating PDF?

Example:

```
Dim strPDF As String
Dim strTxt As String

On Error Resume Next

Screen.MousePointer = vbHourglass

strPDF = App.Path & "\file.pdf"
strTxt = App.Path & "\file.txt"

oPDFmaker.CreatePDFfromTextFile3 strPDF, strTxt

Screen.MousePointer = vbDefault

If Err.Number Then
   MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
   'open PDF file
```

```
End If
```

1.22 CreatePDFfromVisio - Method to create an Adobe Acrobat (PDF) file from a Visio file.

Requirements: Visio

Arguments:

strPDFFile As String	PDF file to create
strVisioFile As String	Visio file to use
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional blnCloseAcrobat As Boolean	Close Acrobat after creating PDF?

Example:

```
oPDFMaker.CreatePDFfromVisio “c:\file.pdf”, “c:\file.vsd”
```

1.23 CreatePDFfromWebPage - Method to create an Adobe Acrobat (PDF) file from a web page.

Requirements: Internet connection and file PDFOSEND.EXE (see page 6 of this document), Acrobat 6 and greater

Arguments:

strURL As String	Web address to use
strPDFFile As String	PDF file to create
Optional blnHideAcrobat As Boolean	Hide Acrobat? Values: True,False
Optional intSecsToWaitForPDF As Integer = 2	Seconds to wait for PDF to complete

Example:

```
Dim strURL As String
Dim strPDF As String
Dim i As Long

On Error Resume Next

strURL = "http://www.microsoft.com"
strPDF = App.Path & "\webpage.pdf"

'Use this awesome function to create a PDF from a web page!

PDFMaker1.CreatePDFfromWebPage strURL, strPDF, True

If Err.Number Then
   MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
   'open PDF file
   oPDFMaker1.OpenPDF strPDF
End If
```

1.24 CreatePDFfromWord - Method to create an Adobe Acrobat (PDF) file from a Word document.

Requirements: Word

Arguments:

strPDFFile As String	PDF file to create
strWordDocument As String	Word document to use
Optional strPassword As String	Word document password
Optional strWritePassword As String	Word document write password
Optional strPages As String	Word document pages to use
Optional blnLandscape As Boolean	Page orientation. Values: True,False
Optional blnPageNumberInFooter As Boolean	Place page number in footer. Values: True,False

Optional blnPageNumberInHeader As Boolean	Place page number in header. Values: True,False
Optional lngPageNumberAlignment As Long	Page number alignment. Values: 0=left, 1=center, 2=right
Optional blnFirstPage As Boolean	Add page number to first page? Values: True,False
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional blnShowWindow As Boolean	Show job process window. Values: True,False
Optional blnSpoolJobs As Boolean	Spool print jobs. Values: True,False
Optional strJobOptionsFile As String	Print job options file.
Optional blnApplySecurity As Boolean	Apply Distiller security settings. Values: True,False

Example:

```
Dim strPDF As String
Dim strDoc As String
Dim strPagesToPrint As Variant

On Error Resume Next

Screen.MousePointer = vbHourglass

strPDF = App.Path & "\debug.pdf"
strDoc = App.Path & "\debug.doc"
strPagesToPrint = "1, 3, 6-8, 10"

oPDFmaker.CreatePDFfromWord strPDF, strDoc, , , strPagesToPrint

Screen.MousePointer = vbDefault

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
  'open PDF file
End If
```

1.25 CreatePDFfromWord2 - Method to create an Adobe Acrobat (PDF) file from a Word document.

Requirements: Word and file PDFOSEND.EXE (see page 6 of this document)

Note: No postscript file created. Creates PDF faster than CreatePDFfromWord()

Arguments:

strPDFFile As String	PDF file to create
strWordDocument As String	Word document to use
Optional strPassword As String	Word document password
Optional strWritePassword As String	Word document write password
Optional strPages As String	Word document pages to use
Optional blnLandscape As Boolean	Page orientation. Values: True,False
Optional blnPageNumberInFooter As Boolean	Place page number in footer. Values: True,False
Optional blnPageNumberInHeader As Boolean	Place page number in header. Values: True,False
Optional lngPageNumberAlignment As Long	Page number alignment. Values: 0=left, 1=center, 2=right
Optional blnFirstPage As Boolean	Add page number to first page? Values: True,False
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional blnCloseAcrobat As Boolean	Close Acrobat after creating PDF file?

Example:

```
Dim strPDF As String
Dim strDoc As String
Dim strPagesToPrint As Variant
On Error Resume Next

Screen.MousePointer = vbHourglass
```

```
strPDF = App.Path & "\debug.pdf"
strDoc = App.Path & "\debug.doc"
strPagesToPrint = "1, 3, 6-8, 10"

oPDFmaker.CreatePDFfromWord2 strPDF, strDoc, , , strPagesToPrint

Screen.MousePointer = vbDefault

If Err.Number Then
   MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
   'open PDF file
End If
```

2. PostScript Creation Methods

2.1 CreatePSfromAccessReport - Method to create a PostScript (PS) file from an Access report.

Requirements: Access

Arguments:

strPSFile As String	Postscript file to create
strDatabase As String	Access database to use
strReportName As String	Report name to use
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional strPassword As String	Access database password
Optional strFilter As String	Report filter to use
Optional strWhereCondition As String	Report where condition to use
Optional strSaveWindowCaption As String	Save window caption Default: "save pdf file as"

2.2 CreatePSfromAutoCAD - Method to create a PostScript (PS) file from an AutoCAD DWG/DXF file.

Requirements: AutoCAD

Arguments:

strPSFile As String	Postscript file to create
strDWGFile As String	DWG/DXF file to use
varLayout As String	Layout name/number to use
Optional strPDFWriter As String	Adobe Acrobat Print Driver. Default: "Acrobat PDFWriter"
Optional blnCloseAcrobat As Boolean	Close Acrobat? Values: True,False
Optional strSaveWindowCaption As String	Save window caption Default: "save pdf file as"

2.3 CreatePSfromCrystalReport - Method to create a PostScript (PS) file from a Crystel report.

Requirements: Crystal Reports

strPSFile As String	Postscript file to create
strCrystalReport As String	Crystal Report to use
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional blnCloseAcrobat As Boolean	Close Acrobat? Values: True,False
Optional strSaveWindowCaption As String	Save window caption Default: "save pdf file as"

2.4 CreatePSfromExcel - Method to create a PostScript (PS) file from an Excel worksheet.

Requirements: Excel

Arguments:

strPSFile As String	Postscript file to create
strExcelWorkbook As String	Excel workbook file to use
VarExcelWorksheet As Variant	Excel worksheet to use. Can be a name or number
strRange As String	Excel worksheet range to use

Optional strPassword As String	Excel workbook password
Optional strWriteResPassword As String	Excel workbook write reserve password
Optional blnUseGrid As Boolean	Use grid lines. Values: True,False
Optional blnLandscape As Boolean	Page orientation. Values: True,False
Optional strHeader As String	Page header
Optional lngHeaderAlignment As Long	Page header alignment. Values: 0=left, 1=center, 2=right
Optional strFooter As String	Page footer
Optional lngFooterAlignment As Long	Page footer alignment. Values: 0=left, 1=center, 2=right
Optional dblTopMargin As Double	Page top margin in inches
Optional dblLeftMargin As Double	Page left margin in inches
Optional dblRightMargin As Double	Page right margin in inches
Optional dblBottomMargin As Double	Page bottom margin in inches
Optional dblHeaderMargin As Double	Page header margin in inches
Optional dblFooterMargin As Double	Page footer margin in inches
Optional lngFirstPageNumber As Long = 1	Beginning page number
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional blnShowWindow As Boolean	Show job process window. Values: True,False
Optional blnSpoolJobs As Boolean	Spool print jobs. Values: True,False

Example:

```
Dim strPS As String
Dim strWB As String
Dim strRange As String

On Error Resume Next

Screen.MousePointer = vbHourglass

strPS = App.Path & "\sample.ps"
strWB = App.Path & "\sample.xls"
strRange = "A1:E276"

oPDFmaker.CreatePSfromExcel strPS, strWB, 1, strRange, , , , , "Page &P", "right", "&D &T", "center", 1#, 0.5, 0#, 1#

Screen.MousePointer = vbDefault

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
  MsgBox "PostScript file created.", vbInformation, "Message"
End If
```

2.5 CreatePSfromMiscFile - Method to create a PostScript (PS) file from a miscellaneous file type.

Requirements: Word

strPSFile As String	Postscript file to create
strMiscFile As String	File to use
Optional strDistiller As String	Adobe Acrobat Distiller Name

2.6 CreatePSfromPowerPoint - Method to create a PostScript (PS) file from a PowerPoint file.

Requirements: Power Point

Arguments:

strPSFile As String	Postscript file to create
strPPDocument As String	PowerPoint file to use
Optional lngStartSlide As Long = -1	PowerPoint start slide to use
Optional lngEndSlide As Long = -1	PowerPoint end slide to use
Optional blnHorizontalOrientation As Boolean	PowerPoint slide orientation. Values: True,False
Optional strDistiller As String	Adobe Acrobat Distiller Name

Optional blnShowWindow As Boolean	Show job process window. Values: True,False
Optional blnSpoolJobs As Boolean	Spool print jobs. Values: True,False

Example:

```
Dim strPS As String
Dim strDoc As String

On Error Resume Next

Screen.MousePointer = vbHourglass

strPS = App.Path & "\file.ps"
strDoc = App.Path & "\file.ppt"

oPDFmaker.CreatePSfromPowerPoint strPS, strDoc

Screen.MousePointer = vbDefault

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
  MsgBox "PostScript file created.", vbInformation, "Message"
End If
```

2.7 CreatePSfromTextFile - Method to create a PostScript (PS) file from a text file.

Requirements: Word

Arguments:

strPSFile As String	Postscript file to create
strTextFile As String	Text file to use
Optional blnLandscape As Boolean	Page orientation. Values: True,False
Optional strFont As String	Font name. Default is “Courier”
Optional blnFontBold As Boolean	Font bold. Values: True,False
Optional intFontSize As Integer = 8	Font size
Optional lngFontColor As Long	Font color
Optional blnFontItalic As Boolean	Font italic. Values: True,False
Optional blnFontUnderline As Boolean	Font underline. Values: True,False
Optional blnPageNumberInFooter As Boolean	Place page number in footer. Values: True,False
Optional blnPageNumberInHeader As Boolean	Place page number in header. Values: True,False
Optional lngPageNumberAlignment As Long	Page number alignment. Values: 0=left, 1=center, 2=right
Optional sngTopMargin As Single	Page top margin
Optional sngLeftMargin As Single	Page left margin
Optional sngRightMargin As Single	Page right margin
Optional sngBottomMargin As Single	Page bottom margin
Optional blnFirstPage As Boolean = True	Add page number to first page? Values: True,False
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional blnShowWindow As Boolean	Show job process window. Values: True,False
Optional blnSpoolJobs As Boolean	Spool print jobs. Values: True,False

Example:

```
Dim strPS As String
Dim strTxt As String

On Error Resume Next

Screen.MousePointer = vbHourglass

strPS = App.Path & "\file.ps"
strTxt = App.Path & "\file.txt"
```

```
oPDFmaker.CreatePSfromTextFile strPS, strTxt, False, "arial", False, 8, 255, False, False, False, True, "right", 20.5, 20.5, , 20.5

Screen.MousePointer = vbDefault

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
  MsgBox "PostScript file created.", vbInformation, "Message"
End If
```

2.8 CreatePSfromTextFile2 - Method to create a PostScript (PS) file from a text file.

Requirements: Word, Notepad

Arguments:

strPSFile As String	Postscript file to create
strTextFile As String	Text file to use
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional strSaveWindowCaption As String	Save window caption Default: "save pdf file as"

2.9 CreatePSfromTextFile3 - Method to create a PostScript (PS) file from a text file.

Arguments:

strPSFile As String	Postscript file to create
strTextFile As String	Text file to use
Optional intRightMargin As Integer	Right margin
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional strSaveWindowCaption As String	Save window caption Default: "save pdf file as"

2.10 CreatePSfromVisio - Method to create a PostScript (PS) file from aVisio file.

Requirements: Visio

Arguments:

strPSFile As String	Postscript file to create
strVisioFile As String	Visio file to use
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional strSaveWindowCaption As String	Save window caption Default: "save pdf file as"

2.11 CreatePSfromWord - Method to create a PostScript (PS) file from a Word document.

Requirements: Word

Arguments:

strPSFile As String	Postscript file to create
strWordDocument As String	Word document to use
Optional strPassword As String	Word document password
Optional strWritePassword As String	Word document write password
Optional strPages As String	Word document pages to use
Optional blnLandscape As Boolean	Page orientation. Values: True,False
Optional blnPageNumberInFooter As Boolean	Place page number in footer. Values: True,False
Optional blnPageNumberInHeader As Boolean	Place page number in header. Values: True,False
Optional lngPageNumberAlignment As Long	Page number alignment. Values: 0=left, 1=center, 2=right
Optional blnFirstPage As Boolean = True	Add page number to first page? Values: True,False
Optional strDistiller As String	Adobe Acrobat Distiller Name
Optional blnShowWindow As Boolean	Show job process window. Values: True,False
Optional blnSpoolJobs As Boolean	Spool print jobs. Values: True,False

Example:

```
Dim strPS As String
Dim strDoc As String
Dim strPagesToPrint As String

On Error Resume Next

Screen.MousePointer = vbHourglass

strPS = App.Path & "\debug.ps"
strDoc = App.Path & "\debug.doc"
strPagesToPrint = "1, 3, 6-8, 10"

oPDFmaker.CreatePSfromWord strPS, strDoc, , , strPagesToPrint

Screen.MousePointer = vbDefault

If Err.Number Then
   MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
   MsgBox "PostScript file created.", vbInformation, "Message"
End If
```

3. Document Methods

3.1 AddScript - Method to add a JavaScript to a document. The script executes when the document is opened.

Arguments:

objDoc As Object	Document object
strName As String	Script name
strScript As String	JavaScript code to execute

3.2 AddThumbnails - Method to add thumbnails to an Adobe Acrobat (PDF) file. See also method RemoveThumbnails.

Arguments:

objDoc As Object	Document object
Optional lngStartPage As Long	Start page number
Optional lngEndPage As Long	End page number

3.3 AddWatermarkFromFile - Method to add a watermark to an Adobe Acrobat (PDF) file from a PDF file.

Requirements: Acrobat 7 and greater.

Arguments:

objDoc As Object	Document object
strSourcePDF As String	PDF file to use as watermark
Optional lngSourcePage As Long	Page in PDF file to use
Optional lngStartPage As Long	Start page number
Optional lngEndPage As Long	End page number
Optional blnOnTop As Boolean = True	Watermark on top?
Optional blnOnScreen As Boolean = True	Display on screen?
Optional blnOnPrint As Boolean = True	Print?
Optional sngScale As Single = 1.0	Size
Optional sngOpacity As Single = 1.0	Transparency level

3.4 AddWatermarkFromText - Method to add a watermark to an Adobe Acrobat (PDF) file from a text string.

Requirements: Acrobat 7 and greater.

Arguments:

objDoc As Object	Document object
strText As String	Text to use as watermark
Optional lngAlignment As Long	Text alignment
Optional lngFont As Long	Text font
Optional intTextSize As Integer = 24	Text font size
Optional varColor As Variant	Text color
Optional lngStartPage As Long	Start page number
Optional lngEndPage As Long	End page number
Optional blnOnTop As Boolean = True	Watermark on top?
Optional blnOnScreen As Boolean = True	Display on screen?
Optional blnOnPrint As Boolean = True	Print?
Optional sngScale As Single = 1.0	Size
Optional sngOpacity As Single = 1.0	Transparency level

3.5 AddWebLinks - Method to add web links to an Adobe Acrobat (PDF) file. See also method RemoveWeblinks.

Arguments:

objDoc As Object	Document object
Optional lngStartPage As Long	Start page number
Optional lngEndPage As Long	End page number

3.6 Calculate - Method to calculate computations stored in Text and ComboBox fields.

Arguments:

objDoc As Object	Document object

3.7 CloseDocObject - Method to close an Adobe Acrobat document.

Arguments:

objDoc As Object	Document object

3.8 CreateDocObject - Method to create a Adobe Acrobat Document object.

Arguments:

ByRef strPDF As String	PDF file to use

Returns: Document Object

3.9 CreateLinksInPDF - Method to create hyperlinks in an Adobe Acrobat (PDF) file.

Requirements: file PDFOSEND.EXE (see page 6 of this document)

Arguments:

strPDFFile As String	PDF file to use
Optional strOpenPassword As String	PDF open password

Example:

```
Dim strPDF As String
Dim i As Integer

strPDF = App.Path & “\invoice.pdf”
'create hyperlinks
oPDFmaker.CreateLinksInPDF strPDF
```

3.10 CreatePageHeaderInPDF - Method to create a page header (page no’s, custom text, date) in an Adobe Acrobat (PDF) file.

Note: From Adobe Acrobat's Document Menu select 'Add Headers & Footers…"

Requirements: file PDFOSEND.EXE (see page 6 of this document)

Arguments:

strPDFFile As String	PDF file to use
intFontName As Integer	Font name (Values: 0 through maximum # in list)
intFontSize As Integer	Font size (Values: 0 through maximum # in list, 0 = 8 pt, 1 = 9 pt, etc)
intPageStyle As Integer	Page style (-1 = don't show page, 0 = #, 1 = 1 of n, 2 = 1/n, 3 = Page #)
ByVal lngPageAlign As Long	Page number alignment (0 = left, 1= center, 2 = right)
intDateStyle As Integer	Date style (-1 = don't show date, 0 = m/d, 1 = m/d/yy, 2 = m/d/yyyy)
ByVal lngDateAlign As Long	Date alignment (0 = left, 1= center, 2 = right)
strCustomText As String	Custom text
ByVal lngCustomTextAlign As Long	Custom text alignment (0 = left, 1= center, 2 = right)
Optional strOpenPassword As String	PDF open password

Example:

```
Dim strPDF As String
Dim strTXT As String

On Error Resume Next

Screen.MousePointer = vbHourglass

strPDF = App.Path & "\file.pdf"
strTXT = App.Path & "\file.txt"

oPDFmaker.CreatePDFfromTextFile3 strPDF, strTXT

oPDFmaker.CreatePageHeaderInPDF strPDF, 1, 1, 4, pdoAlignRight, 4, pdoAlignLeft, "My PDF File", pdoAlignCenter

Screen.MousePointer = vbDefault

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
  'open PDF file
  oPDFmaker.OpenPDF strPDF
End If
```

3.11 CreateTemplate - Method to create a template from a page in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
Optional lngPage As Long	Page number

3.12 DeletePages() – Method to delete pages in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
Optional lngStartPage As Long	Start page number
Optional lngEndPage As Long	End page number

3.13 DeleteSound - Method to delete sound objects in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
strName As String	Sound name to delete

3.14 EmailDocument - Method to email the current document.

Arguments:

objDoc As Object	Document object
Optional blnShowWindow As Boolean = True	Show user interface?
Optional strTo As String	Message recipients
Optional strCC As String	Message carbon copy recipients
Optional strBCC As String	Message blind copy recipients
Optional strSubject As String	Message subject
Optional strBody As String	Message body

3.15 EmailForm - Method to email the current document's form fields as an FDF file.

Arguments:

objDoc As Object	Document object
Optional blnShowWindow As Boolean = True	Show user interface?
Optional strTo As String	Message recipients
Optional strCC As String	Message carbon copy recipients
Optional strBCC As String	Message blind copy recipients
Optional strSubject As String	Message subject
Optional strBody As String	Message body

3.16 ExportAsFDF - Method to export form fields to a FDF file.

Arguments:

objDoc As Object	Document object
Optional blnAllFields As Boolean	Export all fields?
Optional blnNoPassword As Boolean = True	Include text fields with password flag set to True?
Optional varFields As Variant	Array or string of fields to submit
Optional blnFlags As Boolean	Include field flags?
Optional strPath As String	FDF file path
Optional blnAnnotations As Boolean	Include annotations? (Acrobat version 6 and greater)

3.17 ExportAsText - Method to export form fields to a tab-delimited text file.

Arguments:

objDoc As Object	Document object
Optional blnNoPassword As Boolean = True	Include text fields with password flag set to True?
Optional varFields As Variant	Array or string of fields to submit
Optional strPath As String	Text file path

3.18 ExportAsXFDF - Method to export form fields to a XFDF file.

Arguments:

objDoc As Object	Document object
Optional blnAllFields As Boolean	Export all fields?
Optional blnNoPassword As Boolean = True	Include text fields with password flag set to True?
Optional varFields As Variant	Array or string of fields to submit
Optional blnFlags As Boolean	Include field flags?
Optional strPath As String	XFDF file path
Optional blnAnnotations As Boolean	Include annotations? (Acrobat version 6 and greater)

3.19 ExportDataObject - Method to export a specified document object to an external file.

Arguments:

objDoc As Object	Document object

strName As String	Data object name
Optional strPath As String	File path
Optional blnAllowAuthorization As Boolean	Show authorization dialog? (Acrobat version 6 and greater)

3.20 ExportXFAData - Method to export a document's XFA data file to an external file.

Requirements: Acrobat 6 and greater

Arguments:

objDoc As Object	Document object
Optional strPath As String	File path
Optional blnXDP As Boolean = True	Export in XDP format?

3.21 ExtractPages - Method to extract pages from a document and create a new Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
Optional lngStartPage As Long	Start page number
Optional lngEndPage As Long	End page number
Optional strPath As String	File path

3.22 FindTextInPDF - Method to find specified text in an Adobe Acrobat (PDF) file.

Requirements: file PDFOSEND.EXE (see page 6 of this document) if opening a password protected PDF file

Arguments:

strPDFFile As String	PDF file to use
strText As String	Text to search for
Optional blnCaseSensitive As Boolean	Case sensitive search? Values: True,False
Optional blnWholeWordsOnly As Boolean	Search for whole words only? Values: True,False
Optional strOpenPassword As String	PDF open password

Example:

```
Dim strPDF As String
Dim i As Integer

strPDF = App.Path & “\file.pdf”

oPDFmaker.FindTextInPDF strPDF, “Hello World!”, True, True
```

3.23 FlattenPages - Method to convert all annotations within a document to page contents.

Arguments:

objDoc As Object	Document object
Optional lngStartPage As Long	Start page number
Optional lngEndPage As Long	End page number

3.24 GetDocumentProperties - Method to get the properties of an Adobe Acrobat (PDF) document. Note: Use method SetDocumentProperties to set properties.

Arguments:

objDoc As Object	Document object

Returns: document variable type

3.25 GetDocumentTrigger - Method to get a valid document trigger string.

Arguments:

i As Long	Document trigger number

Returns: valid document trigger string

3.26 GetNumPagesPDF - Method to return the number of pages in an Adobe Acrobat (PDF) file.

Requirements: file PDFOSEND.EXE (see page 6 of this document) if opening a password protected PDF file

Arguments:

strPDFFile As String	PDF file to use
Optional strOpenPassword As String	PDF open password

Returns: Long Integer

# = Success, number of pages

Example:

Dim strPDF As String

strPDF = App.Path & “\file.pdf”

MsgBox oPDFmaker.GetNumPagesPDF(strPDF), vbInformation, “Number of Pages”

3.27 GetPageRotation - Method to get the rotation of the specified page.

Arguments:

objDoc As Object	Document object
Optional lngPage As Long	Page number

Returns: Integer

3.28 GetPagesWordCount - Method to get the total number of words in a document/page.

Arguments:

objDoc As Object
Optional lngPage As Long
Optional varPages As Variant

Returns:

Populates varPages array with total # words, returns total words (Long Integer)

3.29 ImportAnFDF - Method to import an FDF file.

Arguments:

objDoc As Object	Document object
Optional strPath As String	File path

3.30 ImportAnXFDF - Method to import an XFDF file.

Arguments:

objDoc As Object	Document object
Optional strPath As String	File path

3.31 ImportTextData - Method that imports a row of data from a text file. Each row must be tab delimited.

Arguments:

objDoc As Object	Document object
Optional strPath As String	File path
Optional lngRow As Long	Row to use

3.32 ImportXFAData - Method to import XFA data.

Arguments:

objDoc As Object	Document object
Optional strPath As String	File path

3.33 InsertPages - Method to insert pages into an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
strSourcePDF As String	Source PDF file to use
Optional lngInsertAfterPage As Long = -1	Page to insert after
Optional lngStartPage As Long	Start page number
Optional lngEndPage As Long	End page number

3.34 MovePage - Method to move a page in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
Optional lngPage As Long	Page number to move
Optional lngMoveAfterPage As Long = -1	Move after page

3.35 NewPage - Method to insert a new blank page into an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
Optional lngInsertAfterPage As Long	Insert after page

3.36 OpenPDF - Method to open an Adobe Acrobat (PDF) file.

Requirements: file PDFOSEND.EXE (see page 6 of this document)

Arguments:

strPDFFile As String	PDF file to open
Optional strBookmark As String	Bookmark to move to
Optional lngGoToPageNo As Long	Page number to jump to
Optional ByVal lngReadType As Long	Values: 0 = None, 1 = Current Page, 2 = Entire Document
Optional strOpenPassword As String	PDF open password

Example:

```
Dim strPDF As String
Dim strBookmark As String

strPDF = App.Path & “\file.pdf”
strBookmark = “Chapter One”

oPDFmaker.OpenPDF strPDF, , 6, 1
```

3.37 PrintDocument - Method to send the document to a printer.

Arguments:

objDoc As Object	Document object
Optional blnShowWindow As Boolean = True	Display user interface window
Optional lngStartPage As Long	Start page number to print
Optional lngEndPage As Long	End page number to print
Optional blnSilent As Boolean	Print silently?
Optional blnShrinkToFit As Boolean	Shrink to fit paper?
Optional blnPrintAsImage As Boolean	Print as image?
Optional blnReverse As Boolean	Reverse pages?
Optional blnAnnotations As Boolean = True	Print annotations?

3.38 PrintPDF - Method to print an Adobe Acrobat (PDF) file.

Requirements: file PDFOSEND.EXE (see page 6 of this document) if opening a password protected PDF file

Arguments:

strPDFFile As String	PDF file to print
Optional strPrinter As String	Printer name to use
Optional blnSilent As Boolean	Show print window? Values: True,False
Optional strOpenPassword As String	PDF open password
Optional lngFirstPageNo As Long	First page number to print
Optional lngLastPageNo As Long	Last page number to print

Example:

```
Dim strPDF As String

strPDF = App.Path & “\file.pdf”

oPDFmaker.PrintPDF strPDF, “HP Laserjet III”, True
```

3.39 PrintPDFinDir - Method to print all specified Adobe Acrobat (PDF) files in a specific directory.

Arguments:

strDirectory As String	Directory to use
Optional strFiles As String = "*.pdf"	PDF files to print (can use wildcard characters)
Optional strPrinter As String	Printer name to use
Optional blnSilent As Boolean	Show print window? Values: True,False
Optional lngSortType As Long	Sort field (0=None,1=File Name,2=File Size,3=File Date)
Optional lngSortOrder As Long	Sort order (0=Ascending, 1=Descending)

Example:

```
oPDFmaker.PrintPDFinDir App.Path, “a*.pdf”, “HP Laserjet III”, True, 1, 0
```

3.40 ReduceFileSize - Method to reduce the size of an Adobe Acrobat (PDF) file.

Requirements: file PDFOSEND.EXE (see page 6 of this document) , Acrobat 6 and greater

Arguments:

strPDFFile As String	PDF file to print
Optional strOpenPassword As String	PDF open password
Optional lngVersion As Long	Acrobat version (0=Version 4, 1=Version 5, 2=Version 6)

Example:

```
‘Reduce file size by making compatible with Acrobat 6 only
oPDFmaker.ReduceFileSize App.Path, “file.pdf”,  , 2
```

3.41 RemoveThumbnails - Method to remove thumbnails from an Adobe Acrobat (PDF) file. See also method AddThumbnails.

Arguments:

objDoc As Object	Document object
Optional lngStartPage As Long	Start page number
Optional lngEndPage As Long	End page number

Returns: Number of deleted thumbnails (Long Integer)

3.42 RemoveWebLinks - Method to remove web links in an Adobe Acrobat (PDF) file. See also method AddWeblinks.

Arguments:

objDoc As Object	Document object
Optional lngStartPage As Long	Start page number
Optional lngEndPage As Long	End page number

Returns: Number of deleted web links (Long Integer)

3.43 ReplacePages - Method to replace pages in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
strSourcePDF As String	Source PDF file to use
Optional lngReplaceStartPage As Long	Replace start page number
Optional lngStartPage As Long	Start page number
Optional lngEndPage As Long	End page number

3.44 SaveAs - Method to save document as a PDF or alternate file type including DOC, TXT, EPS, HTML, JPEG, XML, RTF, XDP, etc.

Arguments:

objDoc As Object	Document object
strPath As String	File path
Optional lngFileType As Long	File type

3.45 SaveFileAs - Method to save an Adobe Acrobat (PDF) file using a different file format (DOC,TXT,HTM,etc).

Requirements: file PDFOSEND.EXE (see page 6 of this document) , Acrobat 6 and greater

Arguments:

strPDFFile As String	PDF file to use
strFile As String	File to create
Optional strOpenPassword As String	PDF open password
Optional ByVal lngFileType As Long	File format to use

Example:

```
Dim strPDF As String
Dim strDOC As String

strPDF = App.Path & "\sample.pdf"
strDOC = App.Path & "\sample.doc"

‘Save PDF file as a Word document
oPDFmaker.SaveFileAs strPDF, strDOC, , pdoDOC
```

3.46 SetDocumentAction - Method to set the action (JavaScript) of a specific document.

Arguments:

objDoc As Object	Document object
lngTrigger As Long	Document trigger type
strAction As String	JavaScript code to execute

3.47 SetDocumentProperties - Method to set the properties of an Adobe Acrobat (PDF) document. Note: Use method GetDocumentProperties to get properties.

Arguments:

objDoc As Object	Document object
doc As Long	Document variable type

3.48 SetOpenPassword - Method to set the open password for a specific Adobe Acrobat (PDF) file.

Requirements: file PDFOSEND.EXE (see page 6 of this document)

Note: This function will not work with PDF files that already have open passwords

Arguments:

strPDFFile As String	PDF file to use
strOpenPassword As String	Open password to use

Example:

oPDFmaker.SetOpenPassword “c:\file.pdf”, “opensezme”

3.49 SetPageAction - Method to set the open/close actions (JavaScript) of a specific document's page.

Arguments:

objDoc As Object	Document object
lngPage As Long	Page number
lngTrigger As Long	Page trigger type
strAction As String	JavaScript code to execute

3.50 SetPageRotations - Method to set the rotation of the specified page.

Arguments:

objDoc As Object	Document object
Optional lngStartPage As Long	Start page number
Optional lngEndPage As Long	End page number
Optional lngRotation As Long	Rotation type. Values: 0, 90, 180, 270

3.51 SetPageView - Method to set the initial page view of an Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
lngPageView As Long	Initial view style (show/hide bookmarks, show thumbnails, etc)

3.52 SpawnPageFromTemplate - Method to spawn a page in the document using the given template.

Arguments:

objDoc As Object	Document object
strTemplate As String	Template name to use
Optional lngPage As Long	Page number
Optional blnRenameFields As Boolean = True	Rename fields?
Optional blnOverlayPage As Boolean = True	Overlay page?

3.53 SpellCheckPDF - Method to spell check an Adobe Acrobat (PDF) file. Results are returned in a one dimensional array.

Arguments:

objDoc As Object	Document object

Optional strDelimiter As String = "^"	Word delimiter
Optional lngPage As Long	Page number

Returns: Variant (array of words and spelling suggestions)

3.54 SubmitFormToURL - Method to submit the current form to a specified URL.

Arguments:

objDoc As Object	Document object
URL As String	URL to submit form to
Optional blnFDF As Boolean = True	Submit as FDF? If false then submit as HTML
Optional blnEmpty As Boolean	Include empty fields?
Optional varFields As Variant	Field list (one dimensional array)
Optional blnGet As Boolean	Get?
Optional blnAnnotations As Boolean	Include annotations?
Optional blnXML As Boolean	XML?
Optional blnIncrementalChanges As Boolean	
Optional blnPDF As Boolean	
Optional blnCanonical As Boolean	
Optional blnExclNonUserAnnots As Boolean	
Optional blnExclFKey As Boolean	
Optional strPassword As String	
Optional blnEmbedForm As Boolean	

4. Bookmark Methods

4.1 BookmarkExists - Method to determine if a bookmark exists in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
strBookmark As String	Bookmark name
Optional blnCaseSensitive As Boolean	Case sensitive?
Optional blnWholeWord As Boolean = True	Whole word?

Returns: Boolean (True if found, False if not found)

4.2 BookmarksToArray - Method to populate a two dimensional array with bookmark data from an Adobe Acrobat (PDF) file.

Arguments:

strPDFFile As String	PDF file to use
varBookmarks As Variant	Array to store bookmark names

Returns: Long Integer (total number of bookmarks)

4.3 BookmarksToPDF - Copy bookmarks from one Adobe Acrobat (PDF) file to another.

Arguments:

strSourcePDF As String	Source PDF file to get bookmarks from
strTargetPDF As String	Target PDF file to copy bookmarks to

Returns: Long Integer (total number of bookmarks copied)

4.4 CreateBookmarkInPDF - Method to create a bookmark in an Adobe Acrobat (PDF) file.

Note: This function compatible with Acrobat 5 and 6 only. Please use newer bookmark functions.

Requirements: file PDFOSEND.EXE (see page 6 of this document) if opening a password protected PDF file

Arguments:

strPDFFile As String	PDF file to use
strBookmarkTitle As String	Bookmark title
Optional blnCaseSensitive As Boolean	Case sensitive search for bookmark
Optional blnWholeWordsOnly As Boolean	Whole word search for bookmark
Optional blnReset As Boolean	Reset search conditions
Optional strOpenPassword As String	PDF open password

Example:

```
Dim strPDF As String
Dim i As Integer

strPDF = App.Path & “\invoice.pdf”
‘create bookmark
oPDFmaker.CreateBookmarkInPDF strPDF, "Introduction", True, True, True
‘open PDF and move to bookmark
oPDFmaker.OpenPDF strPDF, "Introduction"
```

4.5 CreateBookmarksFromArray - Method to create bookmarks in an Adobe Acrobat (PDF) file from an array.

Arguments:

objDoc As Object	Document object
varArray As Variant	Array of bookmark names
Optional blnExpand As Boolean	Expand bookmarks?

Returns: Long Integer (total number of bookmarks created)

4.6 CreateBookmarksFromArray2 - Method to create bookmarks in an Adobe Acrobat (PDF) file from an array.

Arguments:

objDoc As Object	Document object
varArray As Variant	Array of bookmark names
Optional blnExpand As Boolean	Expand bookmarks?

Returns: Long Integer (total number of bookmarks created)

4.7 CreateBookmarksFromArray3 - Method to create bookmarks in an Adobe Acrobat (PDF) file from an array.

Arguments:

objDoc As Object	Document object
varArray As Variant	Array of bookmark names
Optional blnExpand As Boolean	Expand bookmarks?

Returns: Long Integer (total number of bookmarks created)

4.8 CreateBookmarksFromOutline - Method to create bookmarks in an Adobe Acrobat (PDF) file from an outline.

Arguments:

objDoc As Object	Document object
Optional lngStartPage As Long	Start page number
Optional lngEndPage As Long	End page number

Returns: Long Integer (total number of bookmarks created)

4.9 CreateBookmarksFromPages - Method to create bookmarks in an Adobe Acrobat (PDF) file from pages.

Arguments:

objDoc As Object	Document object

Optional strText As String = "Page"	Bookmark partial name

Returns: Long Integer (total number of bookmarks created)

4.10 CreateBookmarksInPDF - Method to create multiple bookmarks in an Adobe Acrobat (PDF) file.

Note: If you need to create multiple bookmarks in a PDF file use this instead of function CreateBookmarkInPDF because it is much more efficient. This function compatible with Acrobat 5 and 6 only. Please use newer bookmark functions.

Requirements: file PDFOSEND.EXE (see page 6 of this document) if opening a password protected PDF file

Arguments:

strPDFFile As String	PDF file to use
varBookmarkTitles As Variant	Bookmark title
Optional blnCaseSensitive As Boolean	Case sensitive search for bookmark
Optional blnWholeWordsOnly As Boolean	Whole word search for bookmark
Optional blnReset As Boolean	Reset search conditions
Optional strOpenPassword As String	PDF open password

Example:

```
Dim strPDF As String
Dim i As Integer
Dim varBookmarks(2) As String

strPDF = App.Path & “\invoice.pdf”
varBookmarks(0) = “Introduction”
varBookmarks(1) = “Body”
varBookmarks(2) = “Conclusion”
‘create multiple bookmarks
oPDFmaker.CreateBookmarksInPDF strPDF, varBookmarks, True, True, True
‘open PDF and move to first bookmark
oPDFmaker.OpenPDF strPDF, "Introduction"
```

4.11 CreateChildBookmark - Method to create child bookmarks in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
strChildBookmark As String	Child bookmark name to create
Optional strBookmark As String	Bookmark to search for
Optional lngPageNumber As Long	Page number
Optional intIndex As Integer	Bookmark index
Optional varColor As Variant	Bookmark color
Optional ByVal lngStyle As Long	Bookmark style
Optional blnCaseSensitive As Boolean	Case sensitive search?
Optional blnWholeWord As Boolean = True	Whole word search?
Optional intX As Integer	X scroll number
Optional intY As Integer	Y scroll number
Optional blnExpand As Boolean	Expand bookmark?

Returns: Integer (1 if child bookmark was ceated)

4.12 ExecuteBookmark - Method to execute a bookmark in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
strBookmark As String	Bookmark name to execute
Optional blnCaseSensitive As Boolean	Case sensitive search?
Optional blnWholeWord As Boolean = True	Whole word search?

Returns: Boolean (True if successful, False if not successful)

4.13 GetBookmarks - Method to retrieve all bookmarks in an Adobe Acrobat (PDF) file and store in an array.

Arguments:

objDoc As Object	Document object
varBookmarks As Variant	Array to store bookmark names

Returns: Long (total number of bookmarks)

4.14 SetBookmarkAction - Method to set a bookmark's action in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
strAction As String	JavaScript code to execute
Optional strBookmark As String	Bookmark name to search for. If blank set action for all bookmarks
Optional blnCaseSensitive As Boolean	Case sensitive search?
Optional blnWholeWord As Boolean = True	Whole word search?
Optional blnSearchAll As Boolean	Search all bookmarks?

Returns: Long Integer (total number of bookmarks changed)

4.15 SetBookmarkColor - Method to set bookmark's color in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
varColor as Variant	Bookmark color (array)
Optional strBookmark As String	Bookmark name to search for. If blank set color for all bookmarks
Optional blnCaseSensitive As Boolean	Case sensitive search?
Optional blnWholeWord As Boolean = True	Whole word search?
Optional blnSearchAll As Boolean	Search all bookmarks?

Returns: Long Integer (total number of bookmarks changed)

4.16 SetBookmarkExpanded - Method to set bookmark as expanded or collapsed in an Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
blnExpand As Boolean	Values: True (expand), False (Collapse)

Returns: Long Integer (total number of bookmarks affected)

4.17 SetBookmarkName - Method to set a bookmark's name in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
strName As String	New bookmark name
Optional strBookmark As String	Bookmark name to search for. If blank set name for all bookmarks
Optional blnCaseSensitive As Boolean	Case sensitive search?
Optional blnWholeWord As Boolean = True	Whole word search?
Optional blnSearchAll As Boolean	Search all bookmarks?

Returns: Long Integer (total number of bookmarks changed)

4.18 SetBookmarkProperty - Method to set a bookmark's text style and/or color by level in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
varColor As Variant	Bookmark color (array)

ByVal lngStyle As Long	Bookmark style
intLevel As Integer	Bookmark level to change

Returns: Long Integer (total number of bookmarks changed)

4.19 SetBookmarkStyle - Method to set a bookmark's text style in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
ByVal lngStyle As Long	Bookmark style
Optional strBookmark As String	Bookmark name to search for. If blank set style for all bookmarks
Optional blnCaseSensitive As Boolean	Case sensitive search?
Optional blnWholeWord As Boolean = True	Whole word search?
Optional blnSearchAll As Boolean	Search all bookmarks?

Returns: Long Integer (total number of bookmarks changed)

5. Field Methods

5.1 ClearItems - Method to clear the items in a combo box or list box.

Arguments:

objDoc As Object	Document object
strName As String	Field name

5.2 CreateButtonField - Method to create a button field in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
strName As String	Field name
lngPage As Long	Page number
varCoords As Variant	Field location coordinates
Optional strText As String	Button caption
Optional lngBorder As Long	Button border type
Optional intButtonAlignX As Integer = 50	Buton X alignment
Optional intButtonAlignY As Integer = 50	Button Y alignment
Optional blnButtonFitBounds As Boolean	Button fit bounds?
Optional lngFieldDisplay As Long	Field display type
Optional varFillColor As Variant	Field background color (array)
Optional intLineWidth As Long	Field line width
Optional blnReadOnly As Boolean	Read only?
Optional varColor As Variant	Field color (array)
Optional strSubmitName As String	Field submit name
Optional varTextColor As Variant	Field foreground color (array)
Optional lngFont As Long	Text font
Optional intTextSize As Integer = 8	Text size
Optional strToolTipText As String	Tool tip text
Optional lngTrigger As Long	Field trigger type
Optional strAction As String	JavaScript code to execute
Optional lngRotation As Long	Field rotation. Values: 0, 90, 180, 270

5.3 CreateCheckBoxField - Method to create a check box field in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
strName As String	Field name
lngPage As Long	Page number
varCoords As Variant	Field location coordinates
varExportValues As Variant	Field export values (array)

Optional lngBorder As Long	Field border type
Optional strDefaultValue As String	Field default value
Optional lngFieldDisplay As Long	Field display type
Optional varFillColor As Variant	Field background color (array)
Optional intLineWidth As Long	Field line width type
Optional blnReadOnly As Boolean	Read only?
Optional blnRequired As Boolean	Required?
Optional varColor As Variant	Field color (array)
Optional strSubmitName As String	Field submit name
Optional varTextColor As Variant	Field foreground color (array)
Optional intTextSize As Integer = 8	Text size
Optional strToolTipText As String	Tool tip text
Optional VarValue As Variant	Field value
Optional lngTrigger As Long	Field trigger type
Optional strAction As String	JavaScript code to execute
Optional lngRotation As Long	Field rotation. Values: 0, 90, 180, 270
Optional lngGlyphStyle As Long	Field glyph style

5.4 CreateComboBoxField - Method to create a combo box field in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
strName As String	Field name
lngPage As Long	Page number
varCoords As Variant	Field location coordinates
varValues As Variant	Field values (array)
Optional lngBorder As Long	Field border type
Optional blnCommitOnSelChange As Boolean	Commit on selection change?
Optional intCalcOrderIndex As Integer	Calculation order index
Optional strDefaultValue As String	Field default value
Optional blnEditable As Boolean	Editable?
Optional lngFieldDisplay As Long	Field display type
Optional blnDoNotSpellCheck As Boolean	Do not spell check?
Optional varFillColor As Variant	Field background color (array)
Optional intLineWidth As Long	Field line width type
Optional blnReadOnly As Boolean	Read only?
Optional blnRequired As Boolean	Required?
Optional varColor As Variant	Field color (array)
Optional strSubmitName As String	Field submit name
Optional varTextColor As Variant	Field foreground color (array)
Optional lngFont As Long	Text font
Optional intTextSize As Integer = 8	Text size
Optional strToolTipText As String	Tool tip text
Optional varValue As Variant	Field value
Optional lngTrigger As Long	Field trigger type
Optional strAction As String	JavaScript code to execute
Optional lngRotation As Long	Field rotation. Values: 0, 90, 180, 270
Optional varExportValues As Variant	Field export values (array)

5.5 CreateListBoxField - Method to create a list box field in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
strName As String	Field name
lngPage As Long	Page number
varCoords As Variant	Field location coordinates
varValues As Variant	Field values (array)
Optional lngBorder As Long	Field border type
Optional blnCommitOnSelChange As Boolean	Commit on selection change?
Optional strDefaultValue As String	Field default value
Optional lngFieldDisplay As Long	Field display type

Optional varFillColor As Variant	Field background color (array)
Optional blnMultipleSelection As Boolean	Multiple selection?
Optional intLineWidth As Long	Field line width type
Optional blnReadOnly As Boolean	Read only?
Optional blnRequired As Boolean	Required?
Optional varColor As Variant	Field color (array)
Optional strSubmitName As String	Field submit name
Optional varTextColor As Variant	Field foreground color (array)
Optional lngFont As Long	Text font
Optional intTextSize As Integer = 8	Text size
Optional strToolTipText As String	Tool tip text
Optional varValue As Variant	Field value
Optional lngTrigger As Long	Field trigger type
Optional strAction As String	JavaScript code to execute
Optional lngRotation As Long	Field rotation. Values: 0, 90, 180, 270
Optional varExportValues As Variant	Field export values (array)

5.6 CreateRadioButtonField - Method to create a radio button field in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
strName As String	Field name
lngPage As Long	Page number
varCoords As Variant	Field location coordinates
varExportValues As Variant	Field export values (array)
Optional lngBorder As Long	Field border type
Optional strDefaultValue As String	Field default value
Optional lngFieldDisplay As Long	Field display type
Optional varFillColor As Variant	Field background color (array)
Optional intLineWidth As Long	Field line width type
Optional blnReadOnly As Boolean	Read only?
Optional blnRequired As Boolean	Required?
Optional varColor As Variant	Field color (array)
Optional strSubmitName As String	Field submit name
Optional varTextColor As Variant	Field foreground color (array)
Optional intTextSize As Integer = 8	Text size
Optional strToolTipText As String	Tool tip text
Optional VarValue As Variant	Field value
Optional blnRadiosInUnison As Boolean	Radio buttons in unison?
Optional lngTrigger As Long	Field trigger type
Optional strAction As String	JavaScript code to execute
Optional lngRotation As Long	Field rotation. Values: 0, 90, 180, 270
Optional lngGlyphStyle As Long	Field glyph style

5.7 CreateSignatureField - Method to create a signature field in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
strName As String	Field name
lngPage As Long	Page number
varCoords As Variant	Field location coordinates
Optional lngBorder As Long	Field border type
Optional lngFieldDisplay As Long	Field display type
Optional varFillColor As Variant	Field background color (array)
Optional intLineWidth As Long	Field line width type
Optional blnReadOnly As Boolean	Read only?
Optional blnRequired As Boolean	Required?
Optional varColor As Variant	Field color (array)
Optional strSubmitName As String	Field submit name
Optional varTextColor As Variant	Field foreground color (array)
Optional intTextSize As Integer = 8	Text size

Optional strToolTipText As String	Tool tip text
Optional VarValue As Variant	Field value
Optional lngTrigger As Long	Field trigger type
Optional strAction As String	JavaScript code to execute
Optional lngRotation As Long	Field rotation. Values: 0, 90, 180, 270

5.8 CreateTextField - Method to create a text field in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
strName As String	Field name
lngPage As Long	Page number
varCoords As Variant	Field location coordinates
Optional lngAlignment As Long	Field alignment type
Optional lngBorder As Long	Field border type
Optional intCharLimit As Integer	Maximum number of characters allowed in field
Optional blnComb As Boolean	Display box for each character?
Optional intCalcOrderIndex As Integer	Calculation order index
Optional strDefaultValue As String	Field default value
Optional lngFieldDisplay As Long	Field display type
Optional blnDoNotSpellCheck As Boolean	Do not spell check?
Optional blnFileSelect As Boolean	Show file selection dialog?
Optional varFillColor As Variant	Field background color (array)
Optional intLineWidth As Long	Field line width type
Optional blnMultiLine As Boolean	Multiple lines?
Optional blnPassword As Boolean	Password field?
Optional blnReadOnly As Boolean	Read only?
Optional blnRequired As Boolean	Required?
Optional varBorderColor As Variant	Field border color (array)
Optional strSubmitName As String	Field submit name
Optional varTextColor As Variant	Field foreground color (array)
Optional lngFont As Long	Text font name
Optional intTextSize As Integer = 8	Text size
Optional strToolTipText As String	Tool tip text
Optional VarValue As Variant	Field value
Optional lngTrigger As Long	Field trigger type
Optional strAction As String	JavaScript code to execute
Optional lngRotation As Long	Field rotation. Values: 0, 90, 180, 270

5.9 DeleteItem - Method to delete an item in a check box or list box.

Arguments:

objDoc As Object	Document object
strName As String	Field name
Optional intIndex As Integer = -1	Item index to remove

5.10 FieldPages - Method to get the page number or an array of page numbers of a specific field.

Arguments:

objDoc As Object	Document object
strName As String	Field name

Returns: Variant (array of page numbers)

5.11 GetButtonCaption - Method to get the caption of a button field.

Arguments:

objDoc As Object	Document object
strName As String	Field name

Returns: String (button caption)

5.12 GetButtonFieldProperties - Method to get the properties of a button field within an Adobe Acrobat (PDF) document. Note: Use method SetButtonFieldProperties to set properties.

Arguments:

objDoc As Object	Document object
strName As String	Field name

Returns: ButtonFieldType

5.13 GetCheckBoxFieldProperties - Method to get the properties of a check box field within an Adobe Acrobat (PDF) document. Note: Use method SetCheckBoxFieldProperties to set properties.

Arguments:

objDoc As Object	Document object
strName As String	Field name

Returns: CheckBoxFieldType

5.14 GetChildFields - Method to get child fields of specified field and store to an array.

Arguments:

objDoc As Object	Document object
strName As String	Field name
varFields As Variant	Array of child field names

Returns: Integer (# of fields)

5.15 GetComboBoxFieldProperties - Method to get the properties of a combo box field within an Adobe Acrobat (PDF) document. Note: Use method SetComboBoxFieldProperties to set properties.

Arguments:

objDoc As Object	Document object
strName As String	Field name

Returns: ComboBoxFieldType

5.16 GetFieldBorder - Method to get a valid field border string.

Arguments:

i As Long	Field border number

Returns: String (valid field border string)

5.17 GetFieldValue - Method to get the value of a specific form field. Use SetFieldValue to set the value.

Arguments:

objDoc As Object	Document object
strName As String	Field name

Returns: Variant (value of field)

5.18 GetGlyphStyle - Method to get a valid glyph style string for a check box or radio button field.

Arguments:

i As Long	Field glyph style number

Returns: String (valid field glyph style string)

5.19 GetItem - Method to get item export value/name in a combo box or list box.

Arguments:

objDoc As Object	Document object
strName As String	Field name
intIndex As Integer	Item index
Optional blnExportValue As Boolean	Get export value?

Returns: Variant (value of field)

5.20 GetListBoxFieldProperties - Method to get the properties of a list box field within an Adobe Acrobat (PDF) document. Note: Use method SetListBoxFieldProperties to set properties.

Arguments:

objDoc As Object	Document object
strName As String	Field name

Returns: ListBoxFieldType

5.21 GetRadioButtonFieldProperties - Method to get the properties of a radio button field within an Adobe Acrobat (PDF) document. Note: Use method SetRadioButtonFieldProperties to set properties.

Arguments:

objDoc As Object	Document object
strName As String	Field name

Returns: RadioButtonFieldType

5.22 GetSignatureFieldProperties - Method to get the properties of a signature field within an Adobe Acrobat (PDF) document. Note: Use method SetSignatureFieldProperties to set properties.

Arguments:

objDoc As Object	Document object
strName As String	Field name

Returns: SignatureFieldType

5.23 GetTextFieldProperties - Method to get the properties of a text field within an Adobe Acrobat (PDF) document. Note: Use method SetTextFieldProperties to set properties.

Arguments:

objDoc As Object	Document object
strName As String	Field name

Returns: TextFieldType

5.24 GetTrigger - Method to get a valid trigger string.

Arguments:

i As Long	Field trigger number

Returns: String (valid field trigger string)

5.25 InsertItem - Method to insert an item in a combo box or list box.

Arguments:

objDoc As Object	Document object
strName As String	Field name
strItem As String	Item value to insert
Optional strExportValue As String	Export value to insert
Optional intIndex As Integer	Item insert index

5.26 IsBoxChecked - Method to determine if a radio button or check box is checked.

Arguments:

objDoc As Object	Document object
strName As String	Field name
intIndex As Integer	Item index

Returns: Boolean (True if checked, False if unchecked)

5.27 IsDefaultChecked - Method to determine if a radio button or check box's default value is checked.

Arguments:

objDoc As Object	Document object
strName As String	Field name
intIndex As Integer	Item index

Returns: Boolean (True if checked, False if unchecked)

5.28 NumItems - Method to get the number of items in a combo box or list box field.

Arguments:

objDoc As Object	Document object
strName As String	Field name

Returns: Integer (# of items)

5.29 RemoveField - Method to remove a field object from an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
strName As String	Field name

5.30 ResetFormFields - Method to reset form fields to their default values.

Arguments:

objDoc As Object	Document object
Optional varFields As Variant	Array of fields to reset

5.31 SetButtonCaption - Method to set the caption of a button field.

Arguments:

objDoc As Object	Document object
strName As String	Field name
strCaption As String	Button caption

5.32 SetButtonFieldProperties - Method to set the properties of a button field within an Adobe Acrobat (PDF) document. Note: Use method GetButtonFieldProperties to get properties.

Arguments:

objDoc As Object	Document object
strName As String	Field name
fldt As ButtonFieldType	Button field type

5.33 SetCheckBox - Method to set the value of a check box field.

Arguments:

objDoc As Object	Document object
strName As String	Field name
intIndex As Integer	Item index
Optional blnChecked As Boolean	Checked?

5.34 SetCheckBoxFieldProperties - Method to set the properties of a check box field within an Adobe Acrobat (PDF) document. Note: Use method GetCheckBoxFieldProperties to get properties.

Arguments:

objDoc As Object	Document object
strName As String	Field name
fldt As CheckBoxFieldType	Check box field type

5.35 SetComboBoxFieldProperties - Method to set the properties of a combo box field within an Adobe Acrobat (PDF) document. Note: Use method GetComboBoxFieldProperties to get properties.

Arguments:

objDoc As Object	Document object
strName As String	Field name
fldt As ComboBoxFieldType	Combo box field type

5.36 SetDefaultValue - Method to set the value of a check box or radio button.

Arguments:

objDoc As Object	Document object
strName As String	Field name
intIndex As Integer	Item index
Optional blnChecked As Boolean	Checked?

5.37 SetFieldAction - Method to set the action of a specific field.

Arguments:

objDoc As Object	Document object
strName As String	Field name
lngTrigger As Long	Field trigger type
strAction As String	JavaScript code to execute

5.38 SetFieldValue - Method to set the value of a specific form field. Use GetFieldValue to get the value.

Arguments:

objDoc As Object	Document object
strName As String	Field name
varValue As Variant	Field value

5.39 SetFocus - Method to set focus on specified field.

Arguments:

objDoc As Object	Document object

strName As String	Field name

5.40 SetItems - Method to set the items for a combo box or list box.

Arguments:

objDoc As Object	Document object
strName As String	Field name
varItems As Variant	Array of items.

5.41 SetListBoxFieldProperties - Method to set the properties of a list box field within an Adobe Acrobat (PDF) document. Note: Use method GetListBoxFieldProperties to get properties.

Arguments:

objDoc As Object	Document object
strName As String	Field name
fldt As ListBoxFieldType	List box field type

5.42 SetRadioButtonFieldProperties - Method to set the properties of a radio button field within an Adobe Acrobat (PDF) document. Note: Use method GetRadioButtonFieldProperties to get properties.

Arguments:

objDoc As Object	Document object
strName As String	Field name
fldt As RadioButtonFieldType	Radio button field type

5.43 SetSignatureFieldProperties - Method to set the properties of a signature field within an Adobe Acrobat (PDF) document. Note: Use method GetSignatureFieldProperties to get properties.
Arguments:

objDoc As Object	Document object
strName As String	Field name
fldt As SignatureFieldType	Signature field type

5.44 SetTextFieldProperties - Method to set the properties of a text field within an Adobe Acrobat (PDF) document. Note: Use method GetTextFieldProperties to get properties.

Arguments:

objDoc As Object	Document object
strName As String	Field name
fldt As TextFieldType	Text field type

6. Annotation Methods

6.1 CreateComment - Method to create a comment in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
lngPage As Long	Page number
strComment As String	Comment
Optional varColor As Variant	Color (array)
Optional strAuthor As String	Author
Optional strSubject As String	Subject
Optional intLeftMargin As Integer	Left margin
Optional intTopMargin As Integer	Top margin
Optional lngNoteIcon As Long	Note icon
Optional strName As String	Annotation name
Optional blnLock As Boolean	Locked?
Optional blnReadOnly As Boolean	Read only?

Optional blnPopupOpen As Boolean	Popup on open?
Optional lngNoteState As Long	Note status
Optional intBorderWidth As Integer = 1	Border width
Optional sngOpacity As Single = 1.0	Opacity level (values: 0.00 to 1.00)

6.2 CreateFreeTextAnnot - Method to create a FreeText annotation in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
lngPage As Long	Page number
varRectangle As Variant	Annotation rectangle coordinates
strText As String	Annotation text
Optional varColor As Variant	Annotation foreground color (array)
Optional strAuthor As String	Author
Optional strSubject As String	Subject
Optional strName As String	Annotation name
Optional blnLock As Boolean	Lock?
Optional blnReadOnly As Boolean	Read only?
Optional lngAlignment As Long	Annotation alignment type
Optional varFillColor As Variant	Annotation background color (array)
Optional lngFont As Long	Text font name
Optional intFontSize As Integer = 8	Text size
Optional intBorderWidth As Integer = 1	Border width
Optional lngRotation As Long	Annotation rotation. Values: 0, 90, 180, 270
Optional sngOpacity As Single = 1.0	Opacity level (values: 0.00 to 1.00)

6.3 CreateTextAnnot - Method to create a Text annotation in an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
lngPage As Long	Page number
strText As String	Text
Optional varColor As Variant	Color (array)
Optional strAuthor As String	Author
Optional strSubject As String	Subject
Optional intLeftMargin As Integer	Left margin
Optional intTopMargin As Integer	Top margin
Optional lngNoteIcon As Long	Note icon
Optional strName As String	Annotation name
Optional blnLock As Boolean	Locked?
Optional blnReadOnly As Boolean	Read only?
Optional blnPopupOpen As Boolean	Popup on open?
Optional lngNoteState As Long	Note status
Optional intBorderWidth As Integer = 1	Border width
Optional sngOpacity As Single = 1.0	Opacity level (values: 0.00 to 1.00)

6.4 FindAndCommentText - Method to find specified text in an Adobe Acrobat (PDF) file and add a comment.

Arguments:

objDoc As Object	Document object
strFindText As String	Text to find
strText As String	Comment text
Optional lngStartPage As Long	Start page number
Optional lngEndPage As Long	End page number
Optional blnCaseSensitive As Boolean	Case sensitive?
Optional lngNoteIcon As Long	Note icon
Optional varColor As Variant	Color (array)
Optional strAuthor As String	Author
Optional strSubject As String	Subject
Optional intBorderWidth As Integer = 1	Border width

Optional blnLock As Boolean	Locked?
Optional blnReadOnly As Boolean	Read only?
Optional lngAlignment As Long	Alignment
Optional lngNoteState As Long	Note status
Optional blnTransparent As Boolean = True	Transparent?

Returns: Long (# of annotations created)

6.5 FindAndOverwriteText - Method to find specified text in an Adobe Acrobat (PDF) file and replace with free text annotation.

Arguments:

objDoc As Object	Document object
strFindText As String	Text to find
strReplaceText As String	Replacement text
Optional lngStartPage As Long	Start page number
Optional lngEndPage As Long	End page number
Optional blnCaseSensitive As Boolean	Case sensitive?
Optional lngFont As Long	Text font
Optional intWidth As Integer	Replacement text width
Optional varColor As Variant	Foreground color (array)
Optional varFillColor As Variant	Background color (array)
Optional blnBorder As Boolean	Border?
Optional intY As Integer	Y position

Returns: Long (# of annotations created)

6.6 GetAnnotations - Method to populate a one dimensional array with annot names within a document.

Arguments:

objDoc As Object	Document object
varAnnots As Variant	Array to store annotation names
Optional lngPage As Long	Page number

Returns: Long (# of annotations)

6.7 GetFreeTextAnnotProperties - Method to get the properties of a free text annotation within an Adobe Acrobat (PDF) document. Note: Use method SetFreeTextAnnotProperties to set properties.

Arguments:

objDoc As Object	Document object
strName As String	Annotation name

Returns: FreeTextAnnotType

6.8 GetNoteIcon - Method to get a valid note icon string.

Arguments:

i As Long	Note icon number

Returns: String (valid note icon)

6.9 GetNoteState - Method to get a valid note status string.

Arguments:

i As Long	Note state number

Returns: String (valid note state)

6.10 GetTextAnnotProperties - Method to get the properties of a text annotation within an Adobe Acrobat (PDF) document. Note: Use method SetTextAnnotProperties to set properties.

Arguments:

objDoc As Object	Document object
strName As String	Annotation name

Returns: TextAnnotType

6.11 RemoveAnnotations - Method to remove annotations from an Adobe Acrobat (PDF) file.

Arguments:

objDoc As Object	Document object
Optional lngPage As Long	Page number

Returns: Long (# of deleted annotations)

6.12 SetFreeTextAnnotProperties - Method to set the properties of a free text annotation within an Adobe Acrobat (PDF) document. Note: Use method GetFreeTextAnnotProperties to get properties.

Arguments:

objDoc As Object	Document object
strName As String	Annotation name
lngPage As Long	Page number
fldt As FreeTextAnnotType	Free text annot properties

6.13 SetTextAnnotProperties - Method to set the properties of a text annotation within an Adobe Acrobat (PDF) document. Note: Use method GetTextAnnotProperties to get properties.

Arguments:

objDoc As Object	Document object
strName As String	Annotation name
lngPage As Long	Page number
fldt As TextAnnotType	Text annot properties

7. Miscellaneous Methods

7.1 AcrobatVersion - Adobe Acrobat version number.

Arguments: None

Returns: String (Acrobat version number)

7.2 CloseAcrobat - Method to close a hidden instance of Adobe Acrobat.

Arguments:

Optional blnWaitUntilClosed As Boolean = True	Wait until Acrobat is closed before moving to next statement?
Optional intTimeout As Integer = 10	Timeout after n seconds while attempting to close Acrobat

7.3 CreateJavaScriptObject - Method to create a JavaScript object from a Document object. Once created all the documents properties and methods are available.

Arguments:

objDoc As Object	Document object

Returns: Object (JavaScript object)

7.4 CreateTextfileFromPDF - Method to create a text file from an Adobe Acrobat (PDF) file.

Requirements: file PDFOSEND.EXE (see page 6 of this document) if opening a password protected PDF file

Arguments:

strPDFFile As String	PDF file to use
strTextfile As String	Text file to create
Optional blnDoubleSpace As Boolean	Use single or double spacing. Values: True,False
Optional strOpenPassword As String	PDF open password

Example:

```
Dim strPDF As String
Dim strTXT As String
On Error Resume Next

Screen.MousePointer = vbHourglass

strPDF = App.Path & "\pdfmaker.pdf"
strTXT = App.Path & "\pdfmaker.txt"

'Create the text file!
oPDFmaker.CreateTextfileFromPDF strPDF, strTXT

Screen.MousePointer = vbDefault

If Err.Number Then
   MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
   'open text file
   ShellExecute 0, vbNullString, strTXT, vbNullString, vbNullString, 1
End If
```

7.5 DatabaseTableToArray - Method to move items from an ODBC compliant database table to an array.

Arguments:

strDSN As String	ODBC Data source name
strSQL As String	SQL statement
varArray As Variant	Array to populate
Optional strUserID As String	User ID
Optional strPassword As String	User password
Optional blnIncludeHeadings As Boolean	Include field headings?

Returns: Integer

 1 = Success
-1 = varArray is not an array
-2 = Unable to establish a connection to the database
-3 = Unable to open recordset
-4 = Unknown error

7.6 DatabaseTableToTextFile - Method to move items from an ODBC compliant database table to a text file.

Arguments:

strDSN As String	ODBC Data source name
strSQL As String	SQL statement
strTextFile As String	Text file to create
Optional strUserID As String	User ID
Optional strPassword As String	User password
Optional blnIncludeHeadings As Boolean	Include field headings?

Optional ByVal strDelimiter As String	Field delimiter
Optional blnQuotes As Boolean	Include quotes?

Returns: Integer

1 = Success
-1= Unable to establish a connection to the database
-2 = Unable to open recordset
-3 = Unknown error

7.7 DownloadFile - Function to download a file from a remote web server.

Requirements: MSINET.OCX

Note: An internet connection must be established before using this function.
Arguments:

inet As Object
strRemoteHost As String
strRemoteDirectory As String
strLocalFile As String
strRemoteFile As String
Optional strUsername As String
Optional strUserPassword As String

Example:

```
'Note: you must modify the remote host settings before this will work
oPDFmaker.DownloadFile Inet1, "microsoft.com", "/mydirectory", "c:\file.txt", "file.txt"
```

7.8 ExcelToOneDArray - Method to copy a an Excel workbook worksheet range to a one dimensional array.

Arguments:

strWorkbook As String	Excel workbook
varWorksheet As Variant	Worksheet name/number
lngStartRow As Long	Start row number
lngStartCol As Long	Start column number
lngEndRow As Long	End row number
lngEndCol As Long	End column number
varArray As Variant	Array to populate
Optional blnCellValue As Boolean	Use cell values?
Optional strPassword As String	Workbook password
Optional strWriteResPassword As String	Workbook write reserve password

Returns: Integer

1 = Success
-1 = Unable to create Excel object
-2 = varArray is not an array
-3 = Unable to open workbook
-4 = Unable to open worksheet
-5 = Unknown error

7.9 ExcelToTwoDArray - Method to copy a an Excel workbook worksheet range to a two dimensional array.

Arguments:

strWorkbook As String	Excel workbook
varWorksheet As Variant	Worksheet name/number
lngStartRow As Long	Start row number
lngStartCol As Long	Start column number
lngEndRow As Long	End row number

lngEndCol As Long	End column number
varArray As Variant	Array to populate
Optional blnCellValue As Boolean	Use cell values?
Optional strPassword As String	Workbook password
Optional strWriteResPassword As String	Workbook write reserve password

Returns: Integer

1 = Success
-1 = Unable to create Excel object
-2 = varArray is not an array
-3 = Unable to open workbook
-4 = Unable to open worksheet
-5 = Unknown error

7.10 GetAlignment - Method to get a valid alignment string.

Arguments:

i As Long	Alignment number

Returns: String (alignment)

7.11 GetFileType - Method to get a valid file type string.

Arguments:

i As Long	File type number

Returns: String (File type)

7.12 GetFont - Method to get a valid font string.

Arguments:

i As Long	Font number

Returns: String (Font)

7.13 IsAcrobatHidden - Method to determine if an instance of Adobe Acrobat is loaded in memory.

Arguments: None

Returns: Boolean (True if found, False if not found)

7.14 OneDArrayToText - Method to move items in a one dimensional array to a string variable.

Arguments:

varArray As Variant	Array
strText As String	Text string to create
strEndOfRowChar As String	End of row character
Optional strEndOfArrayChar As String	End of array character
Optional blnQuotes As Boolean	Surround text with quotes?

Returns: Integer
1 = Success
-1 = varArray is not an array
-2 = Unknown error

7.15 OneDArrayToTextFile - Method to move items in a one dimensional array to a text file.

Arguments:

varArray As Variant	Array
strTextFile As String	Text file to create
Optional blnQuotes As Boolean	Surround text with quotes?

Returns: Integer
 1 = Success
-1 = varArray is not an array
-2 = Unknown error

7.16 PrinterName - Method to get the Adobe Acrobat printer name.

Arguments: None

Returns: String (printer name)

Example:

```
MsgBox oPDFmaker.PrinterName, vbOKOnly, "Adobe Acrobat Distiller Printer Name"
```

7.17 RecordsetToArray - Method to move items from an ADO recordset to an array.

Arguments:

objRecd As Object	ADO recordset object
varArray As Variant	Array to populate
Optional blnIncludeHeadings As Boolean	Include field headings?

Returns: Integer

 1 = Success
-1 = varArray is not an array
-2 = Unknown error

7.18 RecordsetToTextFile - Method to move items from an ADO recordset to a text file.

Arguments:

objRecd As Object	ADO recordset object
strTextFile As String	Text file to create
Optional blnIncludeHeadings As Boolean	Include field headings?
Optional ByVal strDelimiter As String	Field delimiter
Optional blnQuotes As Boolean	Surround fields with quotes?

Returns: Integer

 1 = Success
-1 = Unknown error

7.19 SetColor - Method that sets color arrays for common colors.

Arguments:

lngColor As Long	Color type number
varColor As Variant	Color array to modify

7.20 SetDefPrinter - Method to set the default system printer.

Arguments:

strPrinter As String	Printer to use

Example:

```
oPDFmaker.SetDefPrinter “HP DeskJet 890C”
```

7.21 SortTwoDArray - Method to sort a two dimensional array.

Arguments:

arrArray As Variant	Array to sort
ParamArray SortOrder() As Variant	Sort order. Value: asc or dec

Returns: Integer

1 = Success
-1 = arrArray is not an array
-2 = Invalid Column Reference
-3 = Column Reference Outside Of Array Bounds
-4 = Invalid Sort Order Type
-5 = Unknown error

7.22 TempFilePath() – Get/set the path to store temporary files created by PDF Maker DLL.

Note: The default path is C:\Temp

Arguments:

None

Example:

oPDFMaker.TempFilePath = “C:\MyDir”

7.23 TextFileToArray - Method to move the contents of a text file to an array.

Arguments:

strTextFile As String	Text file
varArray As Variant	Array to populate
Optional ByVal strDelimiter As String	Field delimiter
Optional blnRemoveQuotes As Boolean	Remove quotes surrounding fields?

Returns: Integer

1 = Success
-1 = Unable to open text file
-2 = varArray is not an array
-3 = Unknown error

7.24 TextFileToOneDArray - Method to move the contents of a text file to a one dimensional array.

Arguments:

strTextFile As String	Text file
varArray As Variant	Array to populate

Returns: Integer

1 = Success
-1 = Unable to open text file
-2 = varArray is not an array
-3 = Unknown error

7.25 TwoDArrayToText - Method to move items in a two dimensional array to a string variable.

Arguments:

varArray As Variant	Array

strText As String	Text string to create
strEndOfRowChar As String	End of row character
Optional strEndOfArrayChar As String	End of array character
Optional blnQuotes As Boolean	Surround text with quotes?

Returns: Integer

1 = Success
-1 = varArray is not an array
-2 = Unknown error

7.26 TwoDArrayToTextFile - Method to move items in a two dimensional array to a text file.

Arguments:

varArray As Variant	Array
strTextFile As String	Text file to create
Optional blnQuotes As Boolean	Surround text with quotes?

Returns: Integer
1 = Success
-1 = varArray is not an array
-2 = Unknown error

7.27 UploadFile - Function to copy a local file to a remote web server.

Requirements: MSINET.OCX

Note: An internet connection must be established before using this function.

Arguments:

inet As Object
strRemoteHost As String
strRemoteDirectory As String
strLocalFile As String
strRemoteFile As String
Optional strUsername As String
Optional strUserPassword As String

Example:

```
'Note: you must modify the remote host settings before this will work
oPDFmaker.UploadFile Inet1, "www.microsoft.com", "/mydirectory", App.Path & "\table.htm", "table.htm", "username", "password"
```

7.28 XMLFileToArray - Method to copy items from an XML table file to a two dimensional array.

Arguments:

strXMLFile As String	XML file
varArray As Variant	Array to populate
Optional blnIncludeHeadings As Boolean	Include field headings?

Returns: Integer
1 = Success
-1 = varArray is not an array
-2 = Unable to establish connection
-3 = Unable to open recordset
-4 = Unknown error

7.29 XMLFileToRecordset - Method to copy items from an XML table file to an ADO recordset.

Arguments:

strXMLFile As String	XML file
objRecd As Object	ADO recordset object

Returns: Integer

1 = Success
-1 = Unable to establish connection
-2 = Unknown error