PDF OCX by SkySof Software last updated: 12/5/12

PDF OCX is a powerful ActiveX Control which allows your programs to automatically create Adobe Acrobat PDF files from practically 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 prying eyes! Our professional, talented staff devoted many hours to create PDF OCX and we think you'll agree that by using PDF OCX 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.

Note: PDF OCX requires the full version of Adobe Acrobat 5.0 or greater. When installing Adobe make certain that you also install the Adobe Distiller which is not installed by default.

Important! Before using PDF OCX you must 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 OCX component to your Visual Basic 6.0 project:

1) Load Visual Basic 6.0.
2) From the Project Menu select "Components..."
3) Select component "PDF_OCX" and click the OK Button. The component should now appear in your toolbox.
4) Double-click on the component in the toolbar to add to a Visual Basic form.

To add the PDF OCX component to your VB .NET project:

1) Load Visual Basic .NET. and create a new project
2) With the mouse cursor over the toolbox click the right mouse button to invoke the popup menu.
3) Select "Customize toolbox…"
4) Select the "COM Components" tab.
5) Scroll down the list until you find the PDFOCX.PDF_OCX component. Select it and click the OK Button. The component should now appear in your toolbox in the "Windows Forms" section.
6) Double-click on the component in the toolbar to add to a Visual Basic form.

Distributing PDF OCX with your application

PDF OCX 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 OCX ActiveX control or in their Window's System Directory. The file PDFOSEND.EXE is used by PDF OCX 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 6.0 and greater

Passwords and encryption can be set from the Adobe Acrobat Distiller 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:

```
Dim i As Integer
i = PDFOCX1.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:

```
Dim i As Integer
 i = PDFOCX1.OpenPDF("C:\MyDir\file.pdf", strOpenPassword:="opensezme")
```

Events

FunctionDone – Executes when the function has completed.
FunctionFail – Executes when the function fails. Provides error message.
FunctionStart – Executes when the function starts.

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

AcrobatVersion – Adobe Acrobat version installed on the local machine.
PrinterName – Name of the Adobe Acrobat Distiller.

Functions

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

Requirements: file PDFOSEND.EXE (see page 1 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

Returns: Integer

 1 = Success
-1 = PDF file does not exist
-2 = Unknown error

Example:

```
Dim strPDF As String
Dim i As Integer

strPDF = App.Path & “\invoice.pdf”
‘create bookmark
i = PDFOCX1.CreateBookmarkInPDF(strPDF, "1. Introduction", True, True, True)
‘open PDF and move to bookmark
i = PDFOCX1.OpenPDF(strPDF, "1. Introduction")
```

****

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. This function is much more efficient.

Requirements: file PDFOSEND.EXE (see page 1 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

Returns: Integer

```
 1 = Success
-1 = PDF file does not exist
-2 = Unknown error
```

Example:

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

strPDF = App.Path & “\invoice.pdf”
varBookmarks(0) = “1. Introduction”
varBookmarks(1) = “2. Body”
varBookmarks(2) = “3. Conclusion”
‘create bookmark
i = PDFOCX1.CreateBookmarksInPDF(strPDF, varBookmarks, True, True, True)
‘open PDF and move to first bookmark
i = PDFOCX1.OpenPDF(strPDF, "1. Introduction")
```

****

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

Requirements: file PDFOSEND.EXE (see page 1 of this document)

Arguments:

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

Returns: Integer

1 = Success
-1 = PDF file does not exist
-2 = Unknown error

Example:

```
Dim strPDF As String
Dim i As Integer

strPDF = App.Path & "\invoice.pdf"
'create hyperlinks
i = PDFOCX1.CreateLinksInPDF(strPDF)
```

****

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 1 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
Dim i As Integer

On Error Resume Next

Screen.MousePointer = vbHourglass

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

i = PDFOCX1.CreatePDFfromTextFile3(strPDF, strTXT)

i = PDFOCX1.CreatePageHeaderInPDF(strPDF, 1, 1, 4, pdoAlignRight, 4, pdoAlignLeft, "My PDF File", pdoAlignCenter)

Screen.MousePointer = vbDefault

If i = 1 Then
  'open PDF file
  i = PDFOCX1.OpenPDF(strPDF)
End If
```

****

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

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

Returns: Integer

 1 = Success
-1 = Source file does not exist
-2 = Unknown error

Example:

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

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.

If PDFOCX1.CreatePDF(strFile, strPDF) Then
   'Open the PDF
   i = PDFOCX1.OpenPDF(strPDF)
End If
```

****

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

Requirements: Access and file PDFOSEND.EXE (see page 1 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?

Returns: Integer

 1 = Success
-1 = Access database does not exist
-2 = Unknown error

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"
If PDFOCX1.CreatePDFfromAccessReport(strPDF, strMDB, strRpt) = 1 Then
  'open PDF file
  i = PDFOCX1.OpenPDF(strPDF)
End If

Screen.MousePointer = vbDefault
```

****

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

Requirements: AutoCAD and file PDFOSEND.EXE (see page 1 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?

Returns: Integer

```
 1 = Success
-1 = AutoCAD file does not exist
-2 = Unknown error
```

Example:

```
Dim strPDF As String
Dim strDWG As String
Dim i As Integer

On Error Resume Next
Screen.MousePointer = vbHourglass
strPDF = App.Path & "\test.pdf"
strDWG = App.Path & “\test.dwg”
If PDFOCX1.CreatePDFfromAutoCAD(strPDF, stDWG, “layout1”) = 1 Then
  'open PDF file
  i = PDFOCX1.OpenPDF(strPDF)
End If

Screen.MousePointer = vbDefault
```

****

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

Requirements: Crystal Reports and file PDFOSEND.EXE (see page 1 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?

Returns: Integer

 1 = Success
-1 = Crystal Report File does not exist
-2 = Unknown error

Example:

```
Dim strPDF As String
Dim strRPT As String
Dim i As Integer

On Error Resume Next
Screen.MousePointer = vbHourglass
strPDF = App.Path & "\test.pdf"
strRPT = App.Path & "\test.rpt"
If PDFOCX1.CreatePDFfromCrystalReport(strPDF, strRPT) = 1 Then
   'open PDF file
   i = PDFOCX1.OpenPDF(strPDF)
End If

Screen.MousePointer = vbDefault
```

****

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

Returns: Integer

 1 = Success
-1 = Excel workbook does not exist
-2 = Unknown error

Example:

```
Dim strPDF As String
Dim strWB As String
Dim strRange As String
Dim i As Integer

On Error Resume Next

Screen.MousePointer = vbHourglass

strPDF = App.Path & "\sample.pdf"
strWB = App.Path & "\sample.xls"

'There are three ways to specify the worksheet range. Use only one:
strRange = "October" 'Range name
'strRange = "A1:E111" 'Row and column range
'strRange = "Print_Area" 'Print range

If PDFOCX1.CreatePDFfromExcel(strPDF, strWB, 1, strRange, , , , , "Page &P", pdoAlignRight, "&D &T", pdoAlignCenter, 1#, 0.5,
0#, 1#) = 1 Then
   'open PDF file
   i = PDFOCX1.OpenPDF(strPDF)
   'ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1
End If

Screen.MousePointer = vbDefault
```

****

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

Returns: Integer

```
 1 = Success
-1 = Image file does not exist
-2 = Unknown error
```

Example:

```
Dim strPDF As String
Dim strJPG As String
Dim i As Integer

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"

If PDFOCX1.CreatePDFfromImageFile(strPDF, strJPG, "My PDF file", "Frank Kusluski", "Cute girl", "keywords go here") = 1 Then
   'open PDF file
```

```
  i = PDFOCX1.OpenPDF(strPDF)
  'ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1
End If

Screen.MousePointer = vbDefault
```

****

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

Note: Worksheets must exist in the same workbook.

Requirements: Excel

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

Returns: Integer

```
 1 = Success
-1 = Excel workbook does not exist
-2 = Unknown error
```

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
Dim i As Integer

On Error Resume Next

'The function CreatePDFfromMultiExcel will create a single PDF file from Excel
'worksheets in the same Excel workbook. To create a single PDF file from Excel
'worksheets in various Excel workbooks please use CreatePDFfromMultiPS
```

```
Screen.MousePointer = vbHourglass

strPDF = App.Path & "\sample.pdf"
strWB = App.Path & "\sample.xls"
varWorksheets(0) = 1
varWorksheets(1) = 2
varWorksheets(2) = 3

'There are three ways to specify worksheet ranges

'1) Range names
varRanges(0) = "October"
'varRanges(1) = "November"
'varRanges(2) = "December"
'2) Row and Column ranges
'varRanges(0) = "A1:E111"
varRanges(1) = "A1:E131"
'varRanges(2) = "A1:E46"
'3) Print ranges
'varRanges(0) = "Print_Area"
'varRanges(1) = "Print_Area"
varRanges(2) = "Print_Area"

varLandscape(0) = False
varLandscape(1) = True
varLandscape(2) = False
varFirstPageNumber(0) = 1
varFirstPageNumber(1) = 4
varFirstPageNumber(2) = 8

If PDFOCX1.CreatePDFfromMultiExcel(strPDF, strWB, varWorksheets, varRanges, , , , varLandscape, "Page &P", pdoAlignRight,
"&D &T", pdoAlignCenter, 1#, 0.5, 0#, 1#, , , varFirstPageNumber) = 1 Then
  'open PDF file
  i = PDFOCX1.OpenPDF(strPDF)
  'ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1
End If

Screen.MousePointer = vbDefault
```

****

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)

Returns: Integer

 1 = Success
-1 = Unknown error

Example:

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

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!
i = PDFOCX1.CreatePDFfromMultiImage(strPDF, img)

Screen.MousePointer = vbDefault

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
  'open PDF file
  i = PDFOCX1.OpenPDF(strPDF)
  'ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1
End If
```

****

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)

Returns: Integer

```
 1 = Success
-1 = Unknown error
```

Example:

```
Dim strPDF As String
Dim strPDF1 As String
Dim strPDF2 As String
Dim i As Integer
Dim pdf(1) As String

On Error Resume Next

Screen.MousePointer = vbHourglass

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

'Create PDF1
i = PDFOCX1.CreatePDFfromImageFile(App.Path & "\image1.pdf", App.Path & "\image1.jpg", "My first file", "Frank Kusluski",
"Cute girl 1", "keywords go here")
strPDF1 = App.Path & "\image1.pdf"

'Create PDF2
i = PDFOCX1.CreatePDFfromImageFile(App.Path & "\image2.pdf", App.Path & "\image2.jpg", "My second file", "Frank Kusluski",
"Cute girl 2", "keywords go here")
strPDF2 = App.Path & "\image2.pdf"

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

```
'finally combine all the PDF files into one PDF file!
i = PDFOCX1.CreatePDFfromMultiPDF(strPDF, pdf)

Screen.MousePointer = vbDefault

If i = 1 Then
  'open PDF file
  i = PDFOCX1.OpenPDF(strPDF)
  'ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1
End If
```

****

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)

Returns: Integer

```
 1 = Success
-1 = Unknown error
```

Example:

```
Dim strPDF As String
Dim i As Integer

On Error Resume Next

Screen.MousePointer = vbHourglass

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

'Create the PDF!
i = PDFOCX1.CreatePDFfromMultiPDFinDir(strPDF, App.Path, "i*.pdf")

Screen.MousePointer = vbDefault

If i = 1 Then
  'open PDF file
  i = PDFOCX1.OpenPDF(strPDF)
  'ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1
End If
```

****

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

Returns: Integer

 1 = Success
-1 = Unknown error

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 i As Integer

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
i = PDFOCX1.CreatePSfromPowerPoint(strPS1, strPpt)

strPS2 = App.Path & "\sample.ps"
strWB = App.Path & "\sample.xls"

'There are three ways to specify the worksheet range. Use only one:
strRange = "October" 'Range name
'strRange = "A1:E111" 'Row and column range
'strRange = "Print_Area" 'Print range

'create PostScript file from an Excel workbook file worksheet
If PDFOCX1.CreatePSfromExcel(strPS2, strWB, 1, strRange, , , , , "Page &P", pdoAlignRight, "&D &T", pdoAlignCenter, 1#, 0.5, 0#,
1#, , , 2) <> 1 Then
   Screen.MousePointer = vbDefault
   Exit Sub
End If

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

'create PostScript file from a Word document file
i = PDFOCX1.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
i = PDFOCX1.CreatePSfromTextFile(strPS4, strTXT, False, "arial", False, 12, RGB(0, 0, 255), False, False, False, True,
pdoAlignRight, 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!
i = PDFOCX1.CreatePDFfromMultiPS(strPDF, ps)

Screen.MousePointer = vbDefault

'open PDF file
i = PDFOCX1.OpenPDF(strPDF)
```

****

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

Returns: Integer

```
 1 = Success
-1 = Powerpoint document does not exist
-2 = Unknown error
```

Example:

```
Dim strPDF As String
Dim strDoc As String
Dim i As Integer

On Error Resume Next

Screen.MousePointer = vbHourglass

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

'i = PDFOCX1.CreatePDFfromPowerPoint(strPDF, strDoc)

If PDFOCX1.CreatePDFfromPowerPoint(strPDF, strDoc) = 1 Then
  'open PDF file
  i = PDFOCX1.OpenPDF(strPDF)
  'ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1
End If

Screen.MousePointer = vbDefault
```

****

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

Returns: Integer

 1 = Success
-1 = Postscript file does not exist
-2 = Unknown error

Example:

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

On Error Resume Next

Screen.MousePointer = vbHourglass

strPS = App.Path & "\sample.ps"
strPDF = App.Path & "\sample.pdf"
strWB = App.Path & "\sample.xls"

'There are three ways to specify the worksheet range. Use only one:
strRange = "October" 'Range name
'strRange = "A1:E111" 'Row and column range
'strRange = "Print_Area" 'Print range

'create a PostScript file from Excel workbook worksheet range
If PDFOCX1.CreatePSfromExcel(strPS, strWB, 1, strRange, , , , , "Page &P", pdoAlignRight, "&D &T", pdoAlignCenter, 1#, 0.5, 0#,
1#) = 1 Then

  'create a PDF file from a PostScript file
  i = PDFOCX1.CreatePDFfromPS(strPDF, strPS)

  'open PDF file
  i = PDFOCX1.OpenPDF(strPDF)
End If

Screen.MousePointer = vbDefault
```

****

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

Returns: Integer

 1 = Success
-1 = Text file does not exist
-2 = Unknown error

Example:

```
Dim strPDF As String
Dim strTXT As String
Dim i As Integer

On Error Resume Next

Screen.MousePointer = vbHourglass

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

If PDFOCX1.CreatePDFfromTextFile(strPDF, strTXT, False, "arial", False, 12, RGB(0, 0, 255), False, False, False, True,
pdoAlignRight, 20.5, 20.5, , 20.5) = 1 Then
   'open PDF file
   i = PDFOCX1.OpenPDF(strPDF)
End If

Screen.MousePointer = vbDefault
```

****

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

Requirements: Internet connection and file PDFOSEND.EXE (see page 1 of this document)

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

Returns: Integer

 1 = Success
-1 = Unknown error

Example:

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

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

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

If PDFOCX1.CreatePDFfromWebPage(strURL, strPDF, True) Then
  'Open the PDF
  i = PDFOCX1.OpenPDF(strPDF)
End If
```

****

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

Returns: Integer

 1 = Success
-1 = Word document does not exist
-2 = Unknown error

Example:

```
Dim strPDF As String
Dim strDoc As String
Dim strPagesToPrint As String
Dim i As Integer

On Error Resume Next
```

```
Screen.MousePointer = vbHourglass

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

If PDFOCX1.CreatePDFfromWord(strPDF, strDoc, , , strPagesToPrint) = 1 Then
  'open PDF file
  i = PDFOCX1.OpenPDF(strPDF)
End If

Screen.MousePointer = vbDefault
```

****

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

Returns: Integer

```
 1 = Success
-1 = Excel workbook does not exist
-2 = Unknown error
```

Example:

```
Dim strPS As String
Dim strWB As String
Dim strRange As String
Dim i As Integer

On Error Resume Next

Screen.MousePointer = vbHourglass

strPS = App.Path & "\sample.ps"
```

```
strWB = App.Path & "\sample.xls"

'There are three ways to specify the worksheet range. Use only one:
strRange = "October" 'Range name
'strRange = "A1:E111" 'Row and column range
'strRange = "Print_Area" 'Print range

If PDFOCX1.CreatePSfromExcel(strPS, strWB, 1, strRange, , , , , "Page &P", pdoAlignRight, "&D &T", pdoAlignCenter, 1#, 0.5, 0#,
1#) = 1 Then
   MsgBox "PostScript file created.", vbInformation, "Message"
End If

Screen.MousePointer = vbDefault
```

****

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

Returns: Integer

```
 1 = Success
-1 = Powerpoint file does not exist
-2 = Unknown error
```

Example:

```
Dim strPS As String
Dim strDoc As String
Dim i As Integer

On Error Resume Next

Screen.MousePointer = vbHourglass

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

If PDFOCX1.CreatePSfromPowerPoint(strPS, strDoc) = 1 Then
   MsgBox "PostScript file created.", vbInformation, "Message"
End If

Screen.MousePointer = vbDefault
```

****

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

Returns: Integer

 1 = Success
-1 = Text file does not exist
-2 = Unknown error

Example:

```
Dim strPS As String
Dim strTXT As String
Dim i As Integer

On Error Resume Next

Screen.MousePointer = vbHourglass

strPS = App.Path & "\file.ps"
strTXT = App.Path & "\file.txt"

If PDFOCX1.CreatePSfromTextFile(strPS, strTXT, False, "arial", False, 12, RGB(0, 0, 255), False, False, False, True, pdoAlignRight,
20.5, 20.5, , 20.5) = 1 Then
   MsgBox "PostScript file created.", vbInformation, "Message"
End If

Screen.MousePointer = vbDefault
```

****

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

Returns: Integer

 1 = Success
-1 = Word document does not exist
-2 = Unknown error

Example:

```
Dim strPS As String
Dim strDoc As String
Dim strPagesToPrint As String
Dim i As Integer

On Error Resume Next

Screen.MousePointer = vbHourglass

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

If PDFOCX1.CreatePSfromWord(strPS, strDoc, , , strPagesToPrint) = 1 Then
   MsgBox "PostScript file created.", vbInformation, "Message"
End If

Screen.MousePointer = vbDefault
```

****

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

Requirements: file PDFOSEND.EXE (see page 1 of this document) if opening a password protected PDF file

Returns: Integer

 1 = Success
-1 = PDF file does not exist
-2 = Unknown error

Example:

```
Dim strPDF As String
Dim strTXT As String
Dim i As Integer

On Error Resume Next
```

```
Screen.MousePointer = vbHourglass

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

'Create the text file!
If PDFOCX1.CreateTextfileFromPDF(strPDF, strTXT) = 1 Then
  'open text file
  ShellExecute 0, vbNullString, strTXT, vbNullString, vbNullString, 1
End If

Screen.MousePointer = vbDefault
```

****

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
```

Returns: Integer

```
 1 = Success
-1 = Unknown error
```

Example:

```
Dim i As Integer
'Note: you must modify the remote host settings before this will work
i = PDFOCX1.DownloadFile(Inet1, "microsoft.com", "/mydirectory", "c:\file.txt", "file.txt")
If i = 1 Then
   MsgBox "File downloaded successfully!", vbOKOnly
Else
   MsgBox "Unable to download file!", vbOKOnly
End If
```

****

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

Requirements: file PDFOSEND.EXE (see page 1 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

Returns: Integer

1 = Success
-1 = PDF file does not exist
-2 = Unknown error

Example:

```
Dim strPDF As String
Dim i As Integer

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

i = PDFOCX1.FindTextInPDF(strPDF, "Hello World!", True, True)
```

****

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

Requirements: file PDFOSEND.EXE (see page 1 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
-1 = PDF file does not exist
-2 = Unknown error

Example:

```
Dim strPDF As String

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

MsgBox PDFOCX1.GetNumPagesPDF(strPDF), vbInformation, "Number of Pages"
```

****

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

Requirements: file PDFOSEND.EXE (see page 1 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

Returns: Integer

1 = Success
-1 = PDF file does not exist
-2 = Unknown error

Example:

```
Dim strPDF As String
Dim strBookmark As String
Dim i As Integer

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

i = PDFOCX1.OpenPDF(strPDF, , 6, pdoReadCurrentPage)
```

****

PrinterName() - Method to return the Adobe Acrobat Distiller name.

Arguments: None

Returns: String (printer name)

Example:

```
MsgBox PDFOCX1.PrinterName, vbOKOnly, "Adobe Acrobat Distiller Printer Name"
```

****

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

Requirements: file PDFOSEND.EXE (see page 1 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

Returns: Integer

```
 1 = Success
-1 = PDF file does not exist
-2 = Unknown error
```

Example:

```
Dim strPDF As String
Dim i As Integer

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

i = PDFOCX1.PrintPDF(strPDF, "HP Laserjet III", True)
```

****

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

Example:

Dim i As Integer

i = PDFOCX1.PrintPDFinDir(App.Path, “a*.pdf”, “HP Laserjet III”, True)

****

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

Requirements: file PDFOSEND.EXE (see page 1 of this document) , Acrobat 6.0

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 6.0 only
Dim i As Integer
i = PDFOCX1.ReduceFileSize(App.Path, “file.pdf”, , 2)

****

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

Requirements: file PDFOSEND.EXE (see page 1 of this document) , Acrobat 6.0

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
Dim i As Integer

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

‘Save PDF file as a Word document
i = PDFOCX1.SaveFileAs(strPDF, strDOC, , pdoDOC)

****

SetDefPrinter() - Method to set the default system printer.

Arguments:

strPrinter As String	Printer to use

Example:

PDFOCX1.SetDefPrinter “HP DeskJet 890C”

****

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

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

Requirements: file PDFOSEND.EXE (see page 1 of this document)

Arguments:

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

Example:

```
Dim i As Integer

i = PDFOCX1.SetOpenPassword("c:\file.pdf", "opensezme")
```

****

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
```

Returns: Integer

1 = Success
-1 = Unknown error

Example:

```
Dim i As Integer

'Note: you must modify the remote host settings before this will work
i = PDFOCX1.UploadFile(Inet1, "www.microsoft.com", "/mydirectory", App.Path & "\table.htm", "table.htm", "username",
"password")
If i = 1 Then
   MsgBox "File uploaded successfully!", vbOKOnly
Else
   MsgBox "Unable to upload file!", vbOKOnly
End If
```