

PDF Maker is a powerful ActiveX DLL 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 Maker and we think you'll agree that by using PDF Maker 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 Frank immediately at [kusluski@mail.ic.net](mailto:kusluski@mail.ic.net)

*Note: PDF Maker 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.*

The cost of PDF Maker is only \$49.95 per developer and only \$249.95 for a site license. If you want to customize the PDF Maker functions to your liking the complete Visual Basic 6.0 source code can be purchased for only \$499.95. All upgrades are free and PDF Maker may be distributed with your application royalty free.

Click the link below now to purchase PDF Maker:

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

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

Developer Individual License – only \$44.95

[https://www.paypal.com/xclick/business=kusluski%40mail.ic.net&item\\_name=PDF+Maker+DLL+Single+Developer+License&item\\_number=2&amount=44.95&no\\_note=1&currency\\_code=USD](https://www.paypal.com/xclick/business=kusluski%40mail.ic.net&item_name=PDF+Maker+DLL+Single+Developer+License&item_number=2&amount=44.95&no_note=1&currency_code=USD)

Developer Site License – only \$224.95

[https://www.paypal.com/xclick/business=kusluski%40mail.ic.net&item\\_name=PDF+Maker+Developer+Site+License&item\\_number=6&amount=224.95&no\\_note=1&currency\\_code=USD](https://www.paypal.com/xclick/business=kusluski%40mail.ic.net&item_name=PDF+Maker+Developer+Site+License&item_number=6&amount=224.95&no_note=1&currency_code=USD)

Complete Visual Basic 6.0 Source Code – only \$449.95

[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=449.95&no\\_note=1&currency\\_code=USD](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=449.95&no_note=1&currency_code=USD)

### **Important! Before using PDF Maker 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 reference to your Visual Basic 6.0 project:**

- 1) Load Visual Basic 6.0.
- 2) From the Project Menu select "References..."
- 3) Select reference "PDFmaker" and click the OK Button.

### **To add the PDF Maker 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 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 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 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 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 in your Visual Basic Application**

After adding the PDF Maker 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 place the following statement in the Form\_Load subroutine:

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

#### **Distributing PDF Maker with your application**

PDF Maker 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 ActiveX DLL or in their Window's System Directory. The file PDFOSEND.EXE is used by PDF Maker 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*

Passwords and encryption are set from the Adobe Acrobat Distiller setup screen. To access these features in Acrobat 6.0:

- 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

**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

Example:

```
Dim strPDF As String
```

```
Dim i As Integer
```

```
strPDF = App.Path & “\invoice.pdf”
```

```
‘create bookmark
```

```
oPDFmaker.CreateBookmarkInPDF strPDF, "1. Introduction", True, True, True
```

```
‘open PDF and move to bookmark
```

```
oPDFmaker.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 because it is much more efficient.*

Requirements: file PDFSEND.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

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 multiple bookmarks
oPDFmaker.CreateBookmarksInPDF strPDF, varBookmarks, True, True, True
'open PDF and move to first bookmark
oPDFmaker.OpenPDF strPDF, "1. Introduction"
```

\*\*\*\*\*

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

Requirements: file PDFSEND.EXE (see page 1 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
```

\*\*\*\*\*

**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 PDFSEND.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  
 intDateStyle As Integer  
 ByVal lngDateAlign As Long  
 strCustomText As String  
 ByVal lngCustomTextAlign As Long  
 Optional strOpenPassword As String

Page number alignment (0 = left, 1 = center, 2 = right)  
 Date style (-1 = don't show date, 0 = m/d, 1 = m/d/yy, 2 = m/d/yyyy)  
 Date alignment (0 = left, 1 = center, 2 = right)  
 Custom text  
 Custom text alignment (0 = left, 1 = center, 2 = right)  
 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

\*\*\*\*\*

**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

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

\*\*\*\*\*

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

Requirements: Access and file PDFSEND.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?

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
```

\*\*\*\*\*

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

Requirements: AutoCAD and file PDFSEND.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?

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

\*\*\*\*\*

**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?

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

\*\*\*\*\*

**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
```

\*\*\*\*\*

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

Requirements: Excel and file PDFSEND.EXE (see page 1 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

\*\*\*\*\*

**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

\*\*\*\*\*

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

Requirements: Access and file PDFSEND.EXE (see page 1 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
```

\*\*\*\*\*

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

Requirements: AutoCAD and file PDFSEND.EXE (see page 1 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
```

\*\*\*\*\*

**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  
 Optional dblBottomMargin As Double  
 Optional dblHeaderMargin As Double  
 Optional dblFooterMargin As Double  
 Optional varFirstPageNumber As Variant  
 Optional strDistiller As String  
 Optional blnShowWindow As Boolean  
 Optional blnSpoolJobs As Boolean  
 Optional blnApplySecurity As Boolean

Page right margin in inches  
 Page bottom margin in inches  
 Page header margin in inches  
 Page footer margin in inches  
 Beginning page numbers (array)  
 Adobe Acrobat Distiller Name  
 Show job process window. Values: True,False  
 Spool print jobs. Values: True,False  
 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
```

\*\*\*\*\*

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

Requirements: Excel and file PDFSEND.EXE (see page 1 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  
 Optional strWriteResPassword As String  
 Optional blnUseGrid As Boolean  
 Optional varLandscape As Variant  
 Optional strHeader As String  
 Optional lngHeaderAlignment As Long  
 Optional strFooter As String  
 Optional lngFooterAlignment As Long  
 Optional dblTopMargin As Double  
 Optional dblLeftMargin As Double  
 Optional dblRightMargin As Double  
 Optional dblBottomMargin As Double  
 Optional dblHeaderMargin As Double  
 Optional dblFooterMargin As Double  
 Optional varFirstPageNumber As Variant  
 Optional strDistiller As String  
 Optional blnCloseAcrobat As Boolean

Excel workbook password  
 Excel workbook write reserve password  
 Use grid lines. Values: True,False  
 Page orientation (array). Values: True,False  
 Page header  
 Page header alignment. Values: 0=left, 1=center, 2=right  
 Page footer  
 Page footer alignment. Values: 0=left, 1=center, 2=right  
 Page top margin in inches  
 Page left margin in inches  
 Page right margin in inches  
 Page bottom margin in inches  
 Page header margin in inches  
 Page footer margin in inches  
 Beginning page numbers (array)  
 Adobe Acrobat Distiller Name  
 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
```

\*\*\*\*\*

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

Arguments:

strPDFFile As String  
varImageFiles As Variant

PDF file to create  
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

\*\*\*\*\*

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

Arguments:

strPDFFile As String  
varPDFFiles As Variant

PDF file to create  
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
```

```
*****
```

**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
```

```
*****
```

**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  
Optional blnShowWindow As Boolean  
Optional blnSpoolJobs As Boolean  
Optional strJobOptionsFile As String  
Optional blnApplySecurity As Boolean

Adobe Acrobat Distiller Name  
Show job process window. Values: True,False  
Spool print jobs. Values: True,False  
Print job options file.  
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
```

\*\*\*\*\*

**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
```

\*\*\*\*\*

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

Requirements: Power Point and file PDFSEND.EXE (see page 1 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
```

```
*****
```

**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
```

```
*****
```

**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
```

```
*****
```

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

Requirements: Word and file PDFOSEND.EXE (see page 1 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
```

```
*****
```

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

Requirements: Notepad and file PDFOSEND.EXE (see page 1 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
```

\*\*\*\*\*

**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"
```

\*\*\*\*\*

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

Requirements: Internet connection and file PDFSEND.EXE (see page 1 of this document), Acrobat 6.0

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
```

```
*****
```

**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
```

```
*****
```

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

Requirements: Word and file PDFSEND.EXE (see page 1 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
```

\*\*\*\*\*

**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
```

```
*****
```

**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
```

```
*****
```

**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
```



\*\*\*\*\*

**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
```

\*\*\*\*\*

**CreateTextfileFromPDF()** - Method to create a text file from 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
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

\*\*\*\*\*

**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"

\*\*\*\*\*

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

Requirements: file PDFSEND.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

Example:

Dim strPDF As String

Dim i As Integer

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

```
oPDFmaker.FindTextInPDF strPDF, "Hello World!", True, True
```

```
*****
```

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

Requirements: file PDFSEND.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

Example:

```
Dim strPDF As String
```

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

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

```
*****
```

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

Requirements: file PDFSEND.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

Example:

```
Dim strPDF As String
```

```
Dim strBookmark As String
```

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

```
strBookmark = "Chapter One"
```

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

```
*****
```

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

Arguments: None

Returns: String (printer name)

Example:

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

\*\*\*\*\*

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

Requirements: file PDFSEND.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

Example:

Dim strPDF As String

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

oPDFmaker.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
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

\*\*\*\*\*

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

Requirements: file PDFSEND.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

oPDFmaker.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 PDFSEND.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
```

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

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

\*\*\*\*\*

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

Arguments:

strPrinter As String	Printer to use
----------------------	----------------

Example:

```
oPDFmaker.SetDefPrinter "HP DeskJet 890C"
```

\*\*\*\*\*

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

Requirements: file PDFSEND.EXE (see page 1 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"
```

\*\*\*\*\*

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

*Note: The default path is the Window's System directory. Set this property to another path such as C:\Temp*

Arguments:

None

Example:

```
oPDFMaker.TempFilePath = "C:\Temp"
```

\*\*\*\*\*

**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"