

PDF ActiveX DLL

Author: SkySof Software

Last Updated: 7/4/2007

Before using PDF ActiveX DLL you must install Adobe's PostScript Printer Driver and GhostScript. Both of these programs are free, easy to install, and can be downloaded from the internet. One of the best reasons for using PDF ActiveX DLL is that you can develop applications to produce PDF files which don't require Adobe Acrobat or any other third party software which must be purchased. Also, you can install your application on as many machines as you like without paying expensive royalty fees! PDF ActiveX DLL is a great, low-cost software development solution for large companies that need PDF publishing capabilities.

- 1) First off you will need a print driver to create **PostScript** files. You can download Adobe's free PostScript Print Driver from <http://www.download.com> or <http://www.adobe.com>

To download this print driver from **Download.com** go to:

http://www.download.com/Adobe-Universal-PostScript-Printer-Driver/3000-2116_4-10184850.html?tag=lst-0-1

To download from **Adobe's** web site go to:

<http://www.adobe.com/support/downloads/main.html>

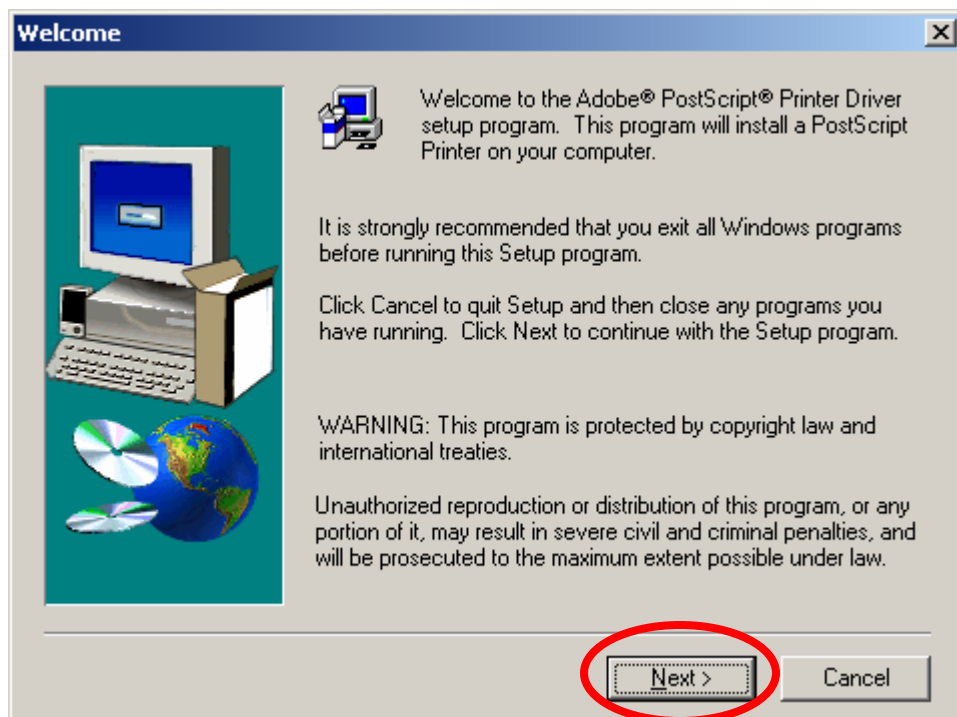
(look for **PostScript printer drivers**)

Alternatively, you can download the Window's version by clicking on this link

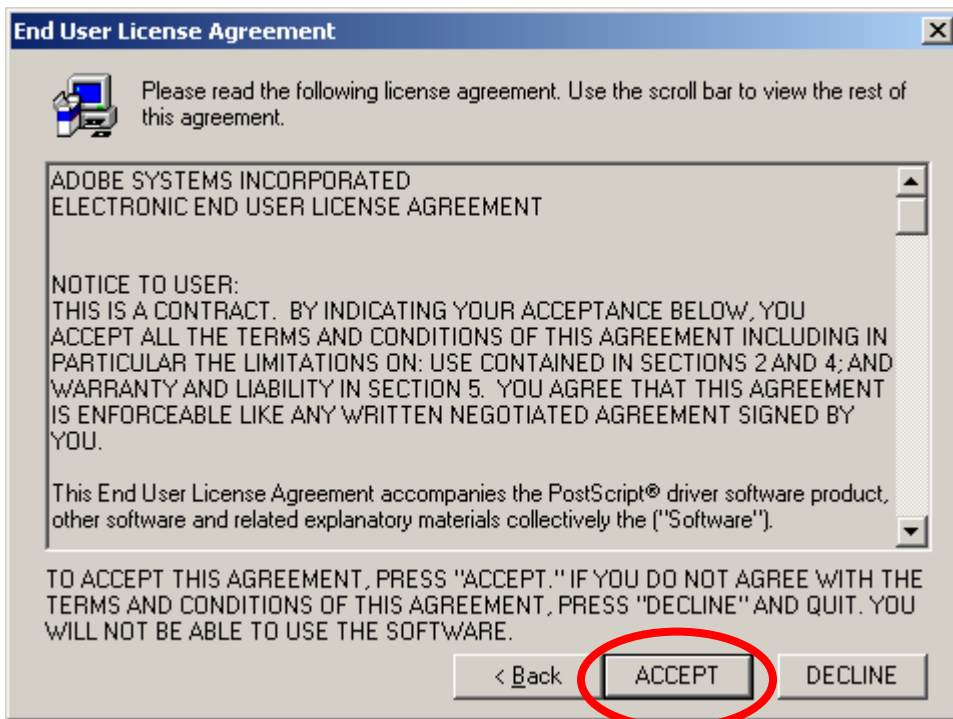
<http://www.getfilez.com/winsteng.exe>

To install the PostScript Print Driver:

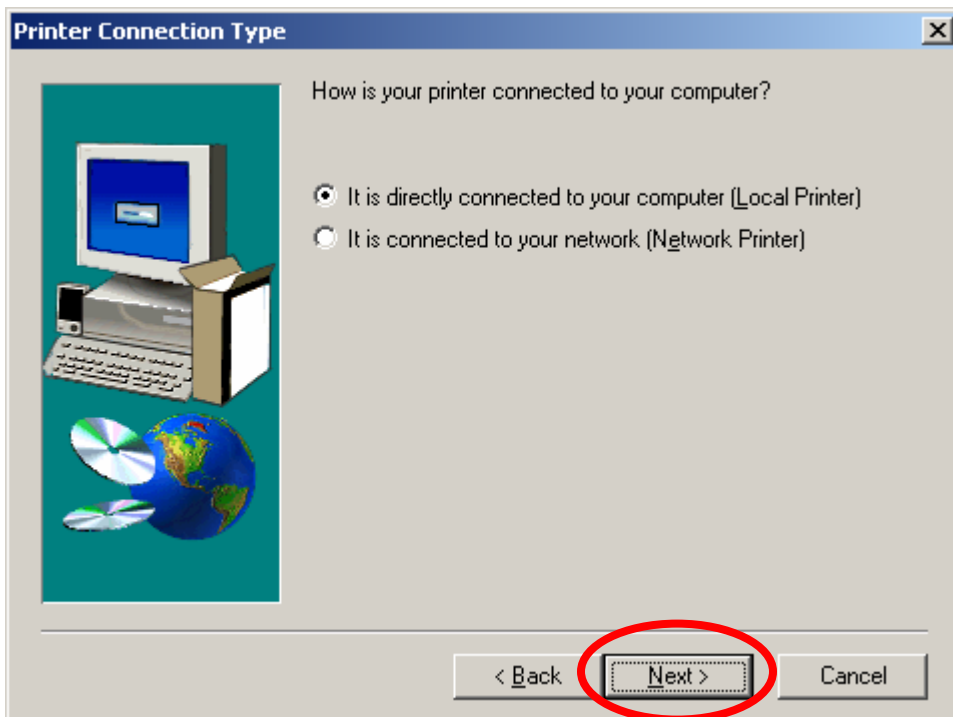
- a) Run the installation file `winsteng.exe`
- b) The following screen will appear. Click the **Next** Button.



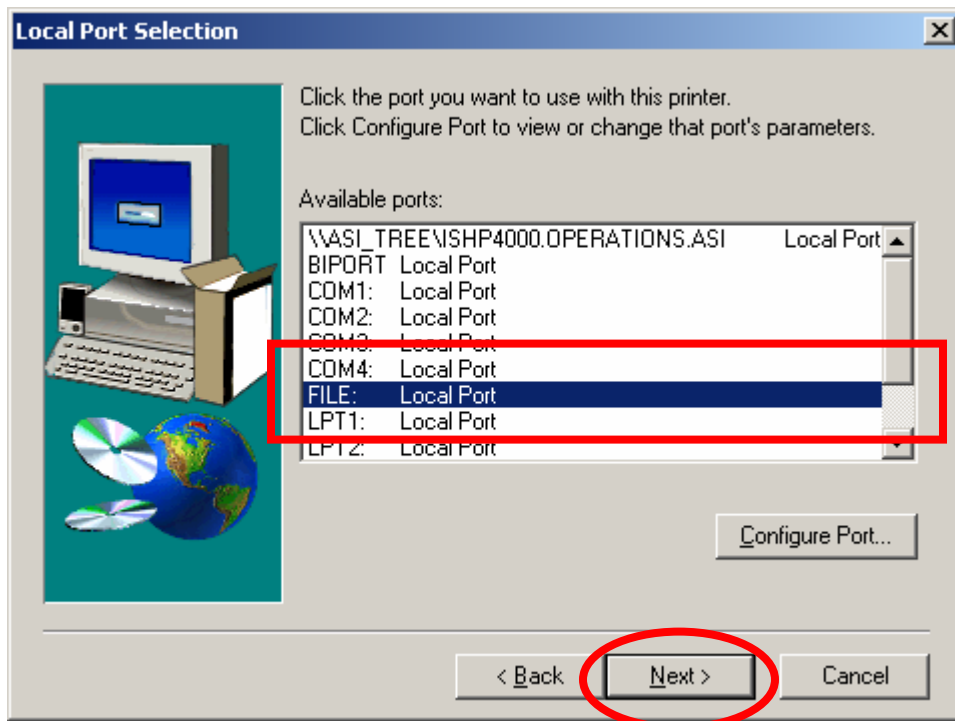
c) Click the **Accept** Button



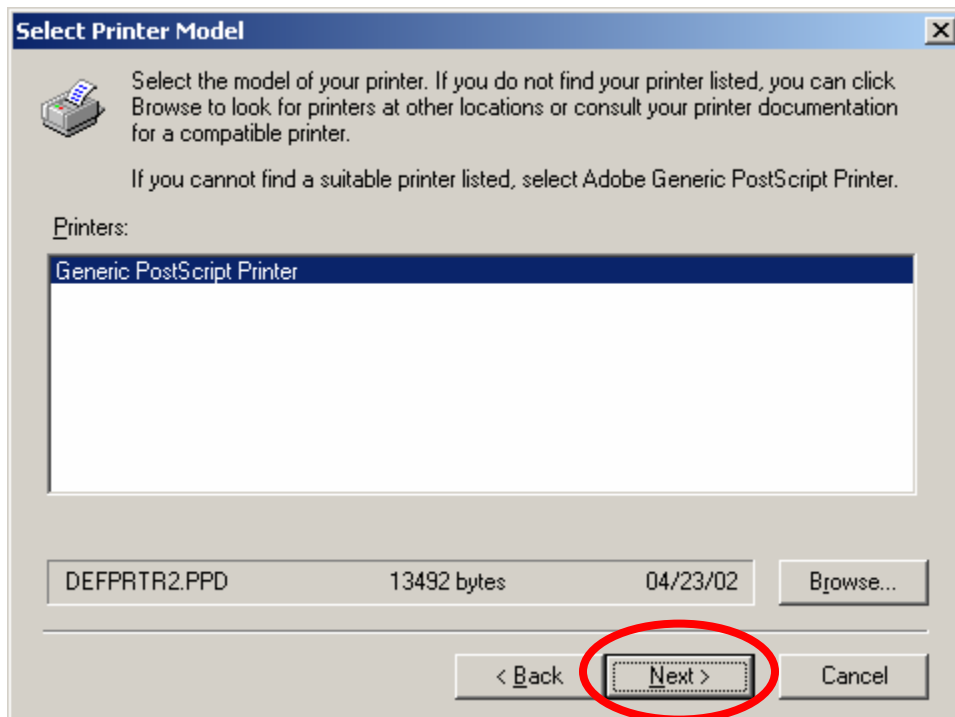
d) Click the **Next** Button



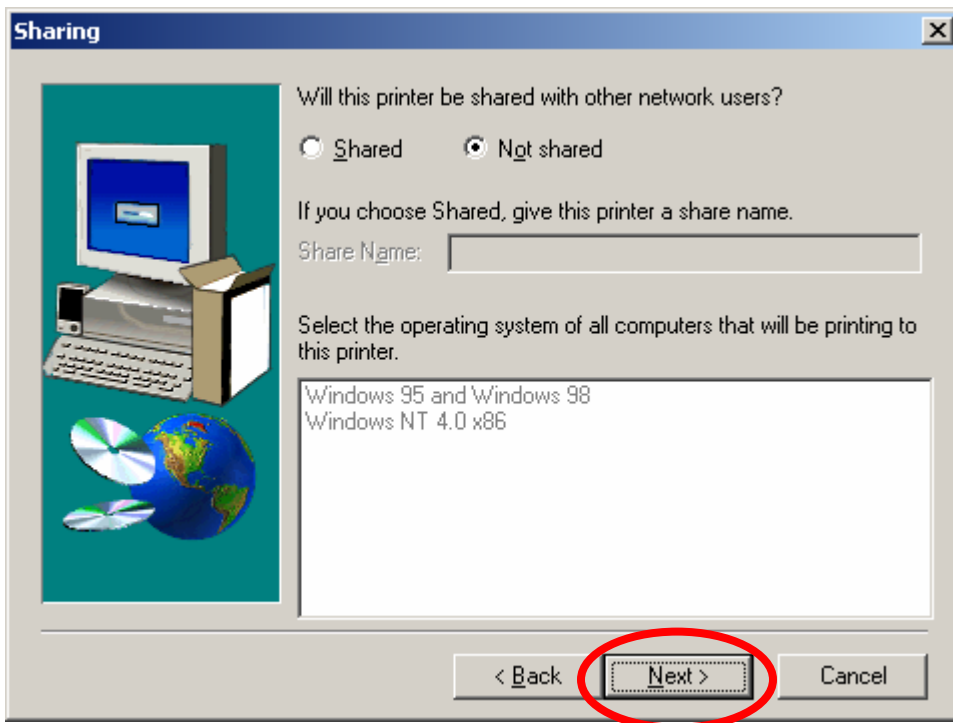
e) Select **FILE:** and click the **Next** Button



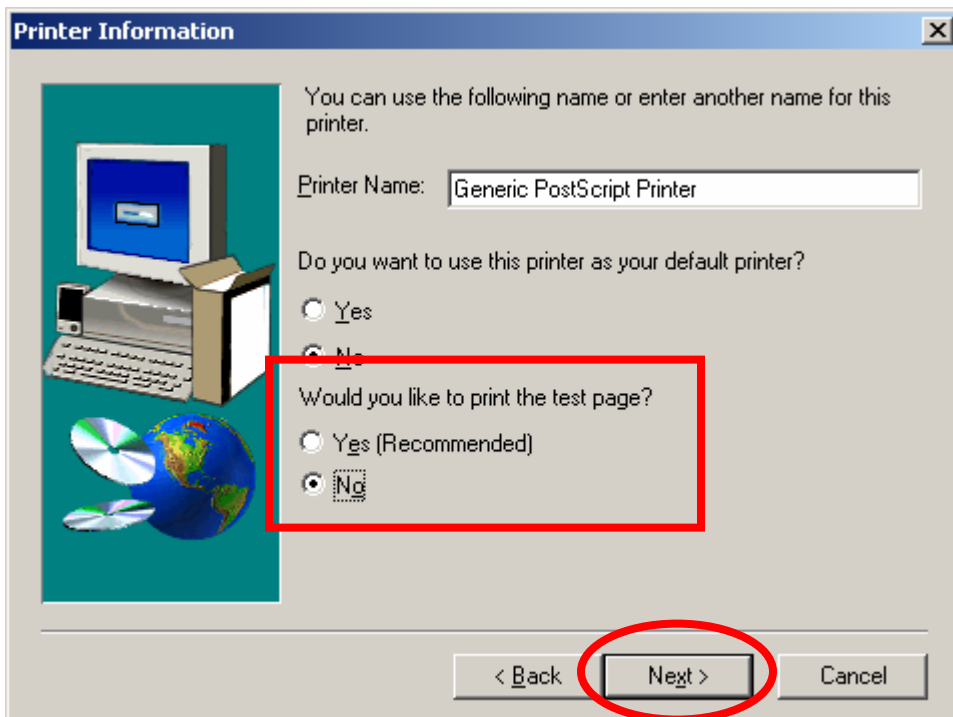
f) Click the **Next** Button



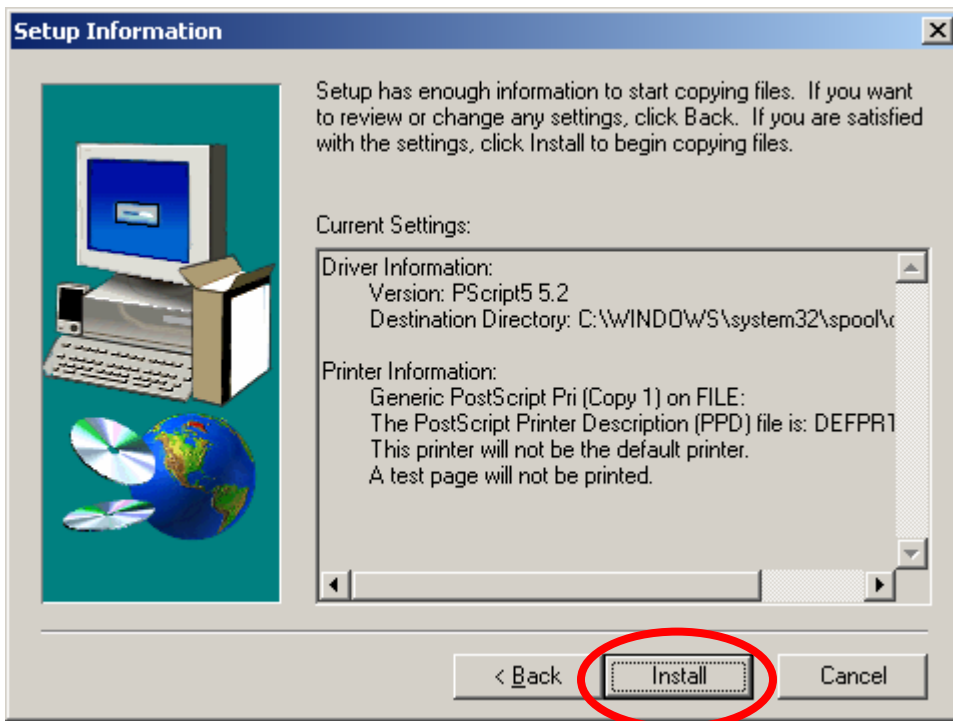
g) Click the **Next** Button



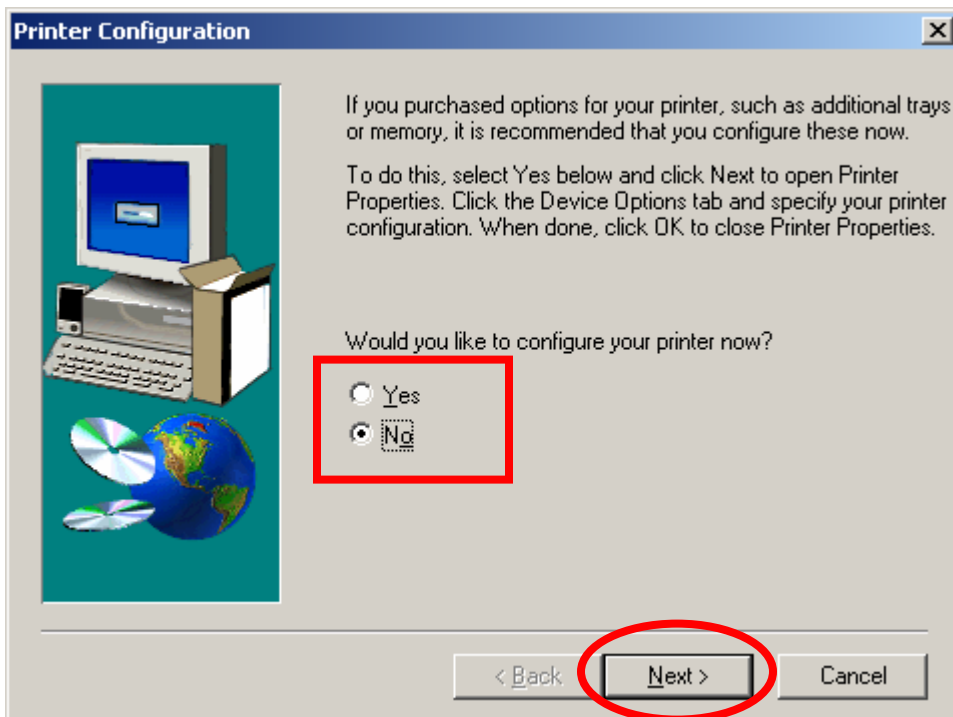
h) Select **No** and click the **Next** Button



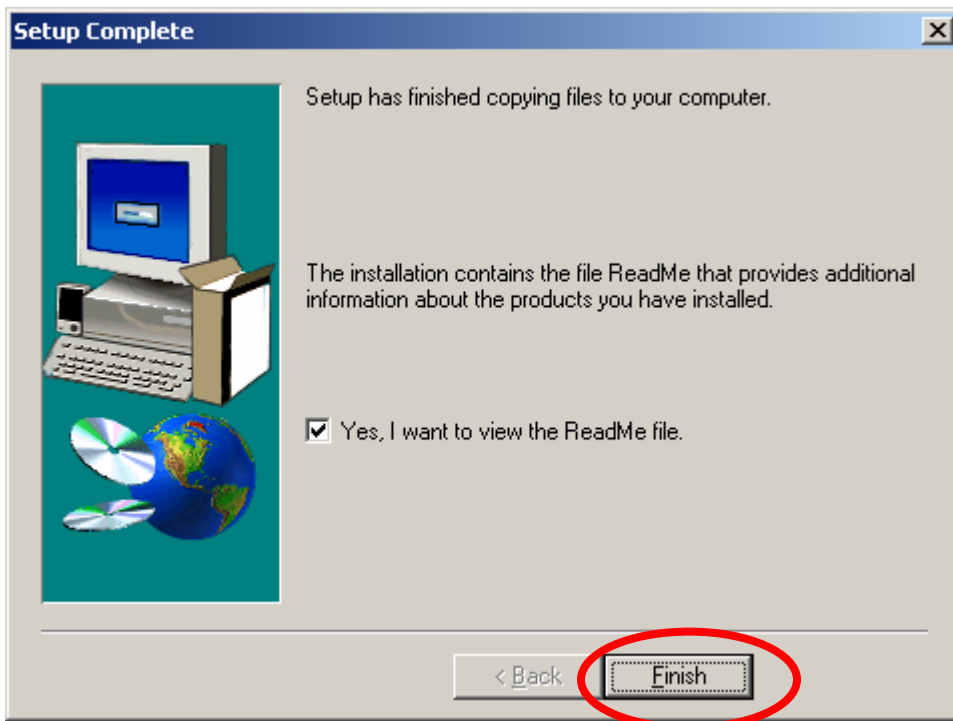
i) Click the **Install** Button



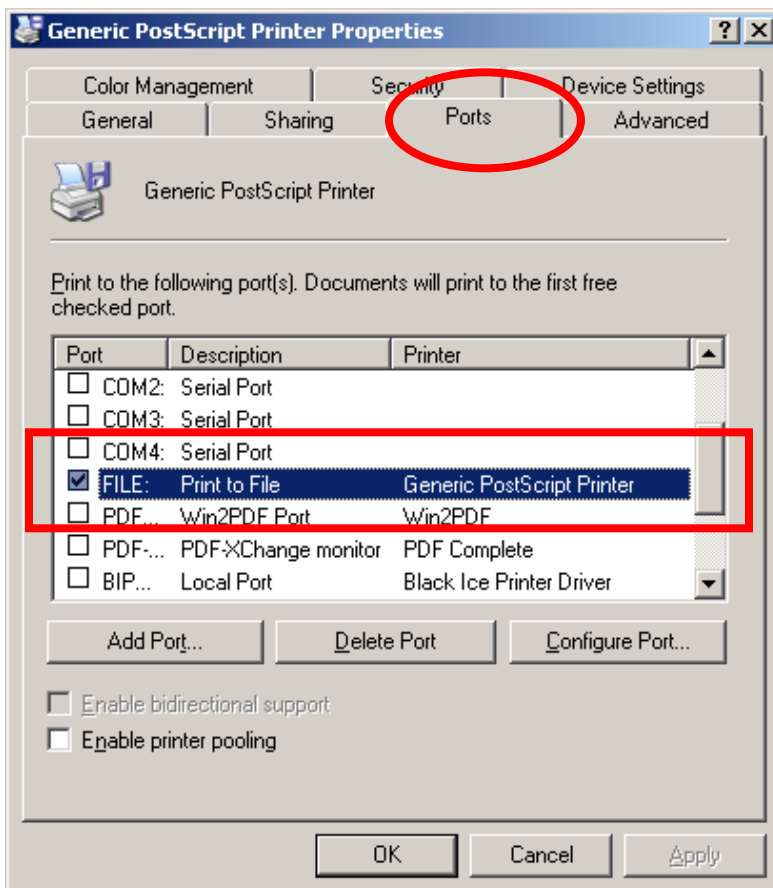
j) Select **No** and click the **Next** Button



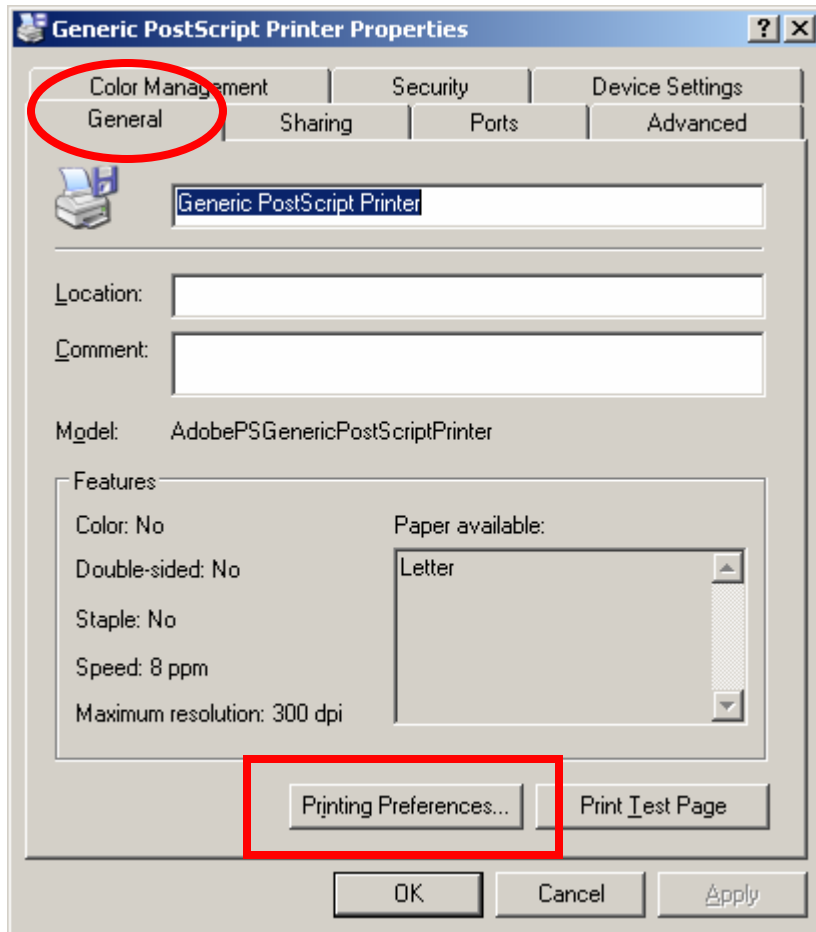
k) Click the **Finish** Button



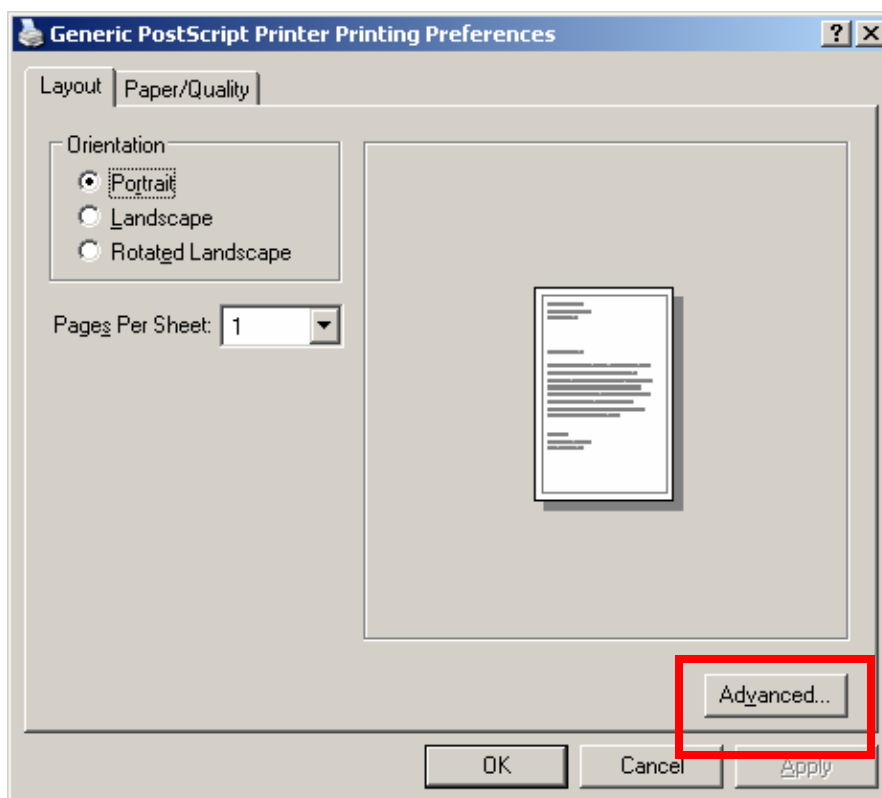
2) After installing the printer driver you will need to modify settings for the PostScript Print Driver. To do so click Window's Start Button, select **Printers and Faxes** from the menu, right-click on **Generic PostScript Printer** and select **Properties** from the popup menu, then click the Ports Tab. Select the **Print to File** option if it is not already selected.



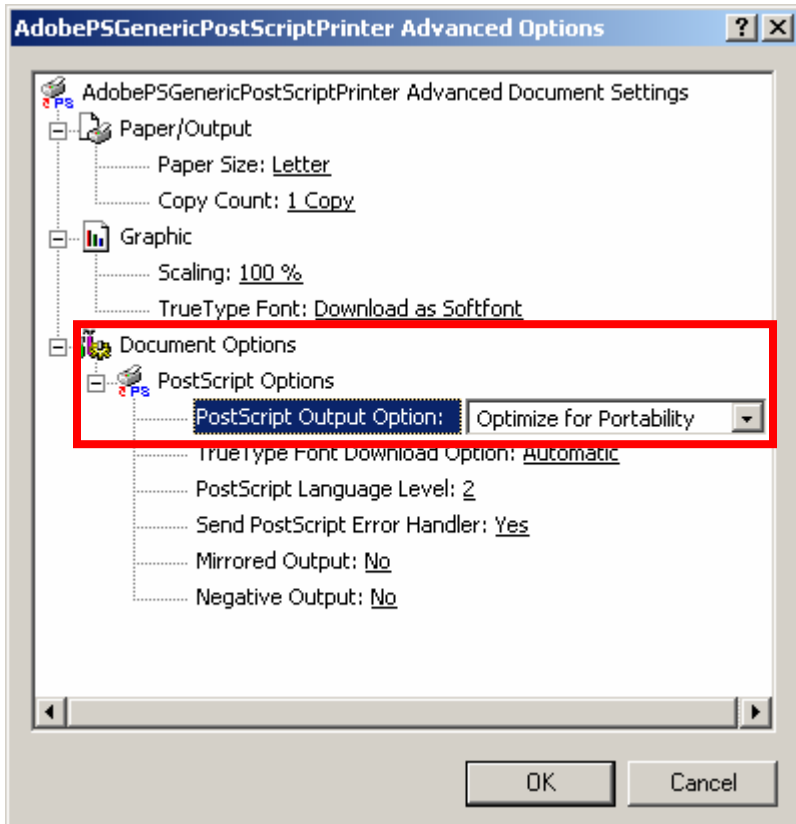
- 3) If you want your PDF files to appear in color you will need to make another change. To do so click the **General Tab** and click the **Printing Preferences** Button.



- 4) Click the **Advanced...** Button



- 5) Change PostScript Output Option to **Optimize for Portability**. If this option is set to **Optimize for Speed** then your PDF documents will appear in black and white. Click the **OK** Buttons to save your changes.



- 6) Congratulations! You finished installing the PostScript Print Driver necessary to create PostScript files! Now you will need to install the free program, **GhostScript**, which converts PostScript files to Portable Document Format files (PDF).

To download GhostScript from the University of Wisconsin Madison's web site go to:

<http://www.cs.wisc.edu/~ghost/doc/AFPL/index.htm>

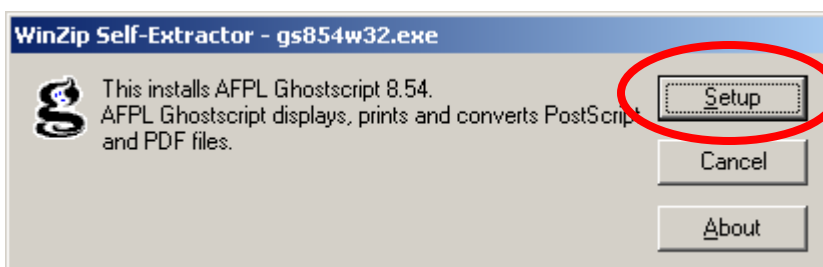
We recommend that you install GhostScript version 8.54 which is the most recent version at the time of this writing. You can download the 32-bit Window's version of GhostScript 8.54 by clicking on this link

<http://www.getfilez.com/g854w32.exe> To download the 64-bit Window's version click here

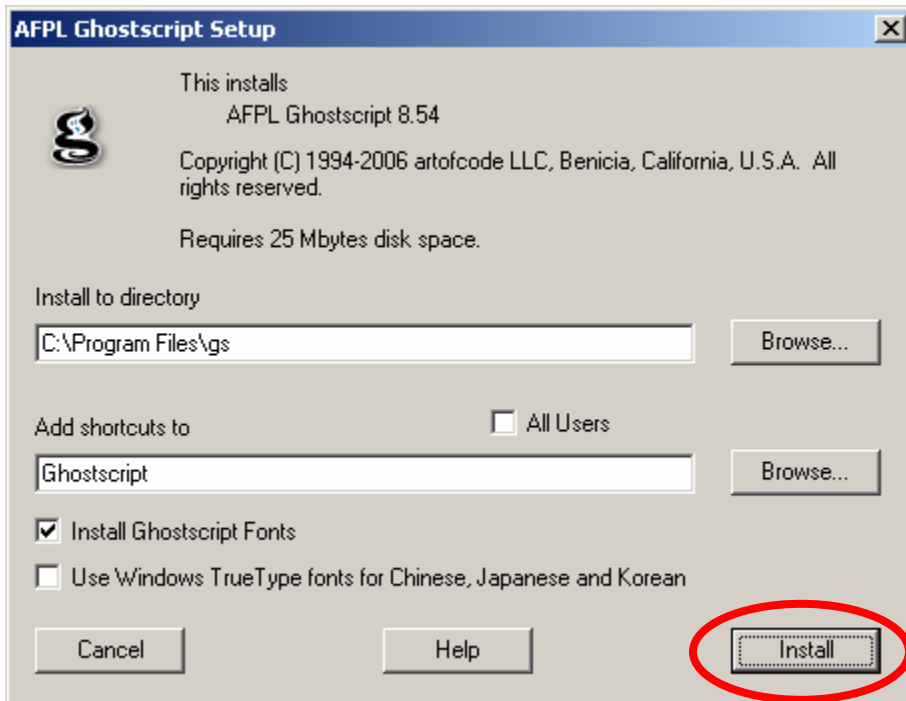
<http://www.getfilez.com/g854w64.exe>

To install GhostScript:

- Run installation file gs854w32.exe or gs854w64.exe
- The following screen will appear. Click the Setup Button.



c) Click the Install Button and wait for the installation to complete.



Using PDF ActiveX DLL with Visual Basic 6.0

- 1) Load Visual Basic 6.0 and create a new Standard EXE
- 2) From the Project Menu select References...
- 3) Select PDFActiveX and click the OK Button
- 4) In the General Declarations section of your form type the following code:
Dim oPDFActiveX As PDFActiveX.CreatePDF

5) In the Form_Load subroutine type:

```
Set oPDFActiveX = New PDFActiveX.CreatePDF  
oPDFActiveX.TempFilePath = "C:\Temp"
```

PDF ActiveX can also be used in your .NET projects, VBA for Excel, VBA for Word, VBA for Access, VBA for PowerPoint, and any other language which is compatible with ActiveX technology.

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

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

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

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

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

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

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

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

Using PDF ActiveX DLL with .NET

- 1) Load .NET and create a new Window's Application
- 2) From the Project Menu select Add Reference...
- 3) Click the COM tab
- 4) Select PDFActiveX in the list and click the Select Button
- 5) Click the OK Button
- 6) At the top of the form type:

Imports PDFActiveX

- 7) Enter the following code after statement Inherits System.Windows.Forms.Form:

Dim oPDFActiveX As PDFActiveX.CreatePDF

- 8) In the Form1_Load subroutine enter code:

oPDFActiveX = New PDFActiveX.CreatePDF
oPDFActiveX.TempFilepath = "C:\Temp"

Distributing PDF ActiveX DLL with your application

PDF ActiveX DLL may be distributed with your application royalty free. When creating the installation program for your application you should include file MSINET.OCX (Microsoft Internet Transfer Control) if you are using the functions UploadFile() and/or DownloadFile(). The file MSINET.OCX can be found in your Window's System directory which is C:\WINDOWS\SYSTEM32 on most machines. Some functions which access the PostScript print driver require 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 ActiveX DLL or in their Window's System Directory. The file PDFOSEND.EXE is used by PDF ActiveX DLL to send messages to the "Print to File" window which pops up when printing with the PostScript print driver.

Properties

PDFPassword – Get/set PDF password (String).

PDFPermissions – Get/set PDF permissions (Long Integer).

PDFVersion – Get/set PDF version (Integer).

TempFilePath – Get/set path where temporary files will be created (String).

GhostScriptVersion – Get GhostScript version number (String).

Functions

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

Requirements: Access and file PDFSEND.EXE

Arguments:

strPDFFile As String	PDF file to create
strDatabase As String	Access database to use
strReportName As String	Report to use
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional strPassword As String	Access database password
Optional strFilter As String	Report filter
Optional strWhereCondition As String	Report where condition
Optional strSaveWindowCaption As Boolean	Window caption. Default: "print to file"

Example:

```
Dim strPDF As String
Dim strMDB As String
Dim strRPT As String
Dim strPrinter As String
On Error Resume Next
```

```
Screen.MousePointer = vbHourglass
strPDF = App.Path & "\Report.pdf"
strMDB = App.Path & "\Author.mdb"
strPrinter = "Generic Postscript Printer"
```

'Note: You may need to modify this file path
'strMDB = "C:\Program Files\Microsoft Visual Studio\VB98\NWIND.MDB"
'If you are using a non-English version of Windows
'you can modify parameter strSaveWindowCaption

```
'Use code below to create a PDF from a single report
strRPT = "authors"
oPDFActiveX.CreatePDFfromAccessReport strPDF, strMDB, strRPT, strPrinter
Screen.MousePointer = vbDefault
If Err.Number Then
    MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
    MsgBox "Access must be installed on this machine.", vbInformation, "Message"
Else
    'open PDF file
    ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1
End If
```

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

Requirements: AutoCAD and file PDFSEND.EXE

Arguments:

strPDFFile As String	PDF file to create
strDWGFile 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 strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional strSaveWindowCaption As Boolean	Window caption. Default: "print to file"

Example:

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

On Error Resume Next

```
strPDF = App.Path & "\HelloWorld.pdf"
strDWG = App.Path & "\HelloWorld.dwg" 'can also use DXF files
strPrinter = "Generic Postscript Printer"
```

```
oPDFActiveX.CreatePDFfromAutoCAD strPDF, strDWG, "layout1", strPrinter
```

If Err.Number Then

```
MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
MsgBox "AutoCAD must be installed on this machine.", vbInformation, "Message"
Else
'open PDF file
ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1
End If
```

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

Requirements: Crystal Reports and file PDFOSEND.EXE

Arguments:

strPDFFile As String	PDF file to create
strCrystalReport As String	Crystal Report File to use
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional strSaveWindowCaption As Boolean	Window caption. Default: "print to file"

Example:

```
Dim strRPT As String
Dim strPDF As String
Dim i As Long
Dim strPrinter As String
```

On Error Resume Next

```
strRPT = App.Path & "\sample.rpt"  
strPDF = App.Path & "\sample.pdf"  
strPrinter = "Generic Postscript Printer"
```

```
oPDFActiveX.CreatePDFfromCrystalReport strPDF, strRPT, strPrinter
```

```
If Err.Number Then
```

```
    MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
```

```
    MsgBox "Crystal Reports must be installed on this machine.", vbInformation, "Message"
```

```
End If
```

```
'Open the PDF
```

```
If Err.Number Then
```

```
    MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
```

```
Else
```

```
    'open PDF file
```

```
    ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1
```

```
End If
```

```
*****
```

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 strPrinter As String	Print driver. Default: "Generic PostScript Printer"
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

Example:

```
Dim strPDF As String
```

```
Dim strWB As String
```

```
Dim strRange As String
Dim strPrinter As String
```

```
On Error Resume Next
```

```
Screen.MousePointer = vbHourglass
```

```
strPDF = App.Path & "\sample.pdf"
strWB = App.Path & "\sample.xls"
'This is the name of the Adobe PostScript Print Driver
strPrinter = "Generic Postscript Printer"
```

```
'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
```

```
oPDFActiveX.CreatePDFfromExcel strPDF, strWB, 1, strRange, strPrinter, , , , "Page &P", pdoAlignRight, "&D
&T", pdoAlignCenter, 1#, 0.5, 0#, 1#
```

```
If Err.Number Then
```

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

```
Screen.MousePointer = vbDefault
```

```
If Err.Number Then
```

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

```
*****
```

CreatePDFfromImageFile() - Method to create an Adobe Acrobat (PDF) file from an image file (JPG,GIF,BMP,TIF,PNG,PCX,etc.). This method does not automatically resize the image to fit the page.

Requirements: File PDFSEND.EXE

Arguments:

strPDFFile As String	PDF file to create
strImageFile As String	Image file to use
Optional sngXpos As Single	X position
Optional sngYpos As Single	Y position
Optional sngWidth As Single	Width in pixels
Optional sngHeight As Single	Height in pixels
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional strSaveWindowCaption As String	Window caption. Default: "print to file"

Example:

```
Dim strPDF As String
Dim strJPG As String
Dim strPrinter As String
```

```
On Error Resume Next
```

```
Screen.MousePointer = vbHourglass
```

```
strPDF = App.Path & "\file.pdf"
strJPG = App.Path & "\image1.jpg"
strPrinter = "Generic Postscript Printer"
```

```
oPDFActiveX.CreatePDFfromImageFile strPDF, strJPG, 100, 1000, 0, 0, strPrinter
'oPDFActiveX.CreatePDFfromImageFile strPDF, strJPG, 100, 1000, 5000, 4000, strPrinter
```

```
If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
End If
```

```
Screen.MousePointer = vbDefault
```

```
If Err.Number = 0 Then
  'open PDF file
  ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1
End If
```

```
*****
```

CreatePDFfromImageFile2() - Method to create an Adobe Acrobat (PDF) file from an image file (JPG,GIF,BMP,TIF,PNG,PCX,etc.). This method automatically resizes the image to fit the page using Window's file RNDLL32.EXE.

Requirements: File PDFSEND.EXE

Arguments:

strPDFFile As String	PDF file to create
strImageFile As String	Image file to use
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional strSaveWindowCaption As String	Window caption. Default: "print to file"

Example:

```
Dim strPDF As String
Dim strJPG As String
Dim strPrinter As String
```

```
On Error Resume Next
```

```
Screen.MousePointer = vbHourglass
```

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

```
strJPG = App.Path & "\image1.jpg"  
strPrinter = "Generic Postscript Printer"
```

```
oPDFActiveX.CreatePDFfromImageFile2 strPDF, strJPG, strPrinter
```

```
If Err.Number Then
```

```
MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"  
End If
```

```
Screen.MousePointer = vbDefault
```

```
If Err.Number = 0 Then
```

```
'open PDF file  
ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1  
End If
```

```
*****
```

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

Requirements: Access and file PDFOSEND.EXE

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 strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional strPassword As String	Password to use
Optional strSaveWindowCaption As String	Window caption. Default: "print to file"

Example:

```
Dim strPDF As String  
Dim strMDB As String  
Dim strRPT As String  
Dim arr(1) As String  
On Error Resume Next  
Screen.MousePointer = vbHourglass  
strPDF = App.Path & "\Report.pdf"  
strMDB = App.Path & "\Author.mdb"  
arr(0) = "Authors"  
arr(1) = "Sales"  
'Note: You may need to modify this file path  
'strMDB = "C:\Program Files\Microsoft Visual Studio\VB98\NWIND.MDB"  
'arr(0) = "invoice"  
'arr(1) = "catalog"  
'If you are using a non-English version of Windows/Acrobat  
'you can modify parameter strSaveWindowCaption  
oPDFActiveX.CreatePDFfromMultiAccessReport strPDF, strMDB, arr  
Screen.MousePointer = vbDefault
```



```

If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
  MsgBox "Access must be installed on this machine.", vbInformation, "Message"
Else
  ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1
End If

```

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

Requirements: AutoCAD and file PDFSEND.EXE

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 strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional strSaveWindowCaption As String	Window caption. Default: "print to file"

Example:

```

Dim arr(1) As String
arr(0) = "Layout1"
arr(1) = "Layout2"
oPDFActiveX.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 strPrinter As String	Print driver. Default: "Generic PostScript Printer"
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)

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 strPassword As String
Dim strPrinter As String
```

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
strPrinter = "Generic Postscript Printer"
```

"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
```

```
oPDFActiveX.CreatePDFfromMultiExcel strPDF, strWB, varWorksheets, varRanges, strPrinter, , , , varLandscape, "Page &P", pdoAlignRight, "&D &T", pdoAlignCenter, 1#, 0.5, 0#, 1#, , , varFirstPageNumber
```

```
If Err.Number Then
```

```
MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
```

```
MsgBox "Excel must be installed on this machine.", vbInformation, "Message"
```

```
End If
```

```
Screen.MousePointer = vbDefault
```

```
If Err.Number Then
```

```
MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
```

```
Else
```

```
'open PDF file
```

```
ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1
```

```
End If
```

```
*****
```

CreatePDFfromMultiPDF() - Method to create a PDF file from multiple PDF files.

Arguments:

strPDFFile As String

PDF file to create

varPDFFiles As Variant

PDF files to use (stored in one dimensional array)

Example:

```
Dim strPDF As String
```

```
Dim strPDFS(1) As String
```

```
On Error Resume Next
```

```
Screen.MousePointer = vbHourglass
```

```
'PDF file to create
```

```
strPDF = AppPath & "\multiple.pdf"
```

```
'PDF files to create
```

```
strPDFS(0) = AppPath & "\file1.pdf"
```

```
strPDFS(1) = AppPath & "\file2.pdf"
```

```
oPDFmaker.CreatePDFfromWord strPDFS(0), AppPath & "\debug.doc", strPrinter
```

```
oPDFmaker.CreatePDFfromExcel strPDFS(1), AppPath & "\sample.xls", 1, "October", strPrinter, , , , "Page &P", pdoAlignRight, "&D &T", pdoAlignCenter, 1#, 0.5, 0#, 1#
```

```
'Combine multiple PDF files to create a single PDF file
```

```
oPDFmaker.CreatePDFfromMultiPDF strPDF, strPDFS
```

```
Screen.MousePointer = vbDefault
```

```
If Err.Number Then
```

```
MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
```

```
Else
  'open PDF file
  ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1
End If
```

CreatePDFfromMultiPDFinDir() - Method to create a PDF file from multiple PDF files.

Arguments:

strPDFFile As String	PDF file to create
strDirectory As String	Directory to scan
Optional strFiles As String = "*.pdf"	PDF files to use (Can use * and ? wildcard characters)
Optional lngSortType As Long	Sort type. Values: 0 = None, 1 = Name, 2 = Size, 3 = Date
Optional lngSortOrder As Long	Sort order. Values: 0 = Ascending, 1 = Descending

Example:

```
Dim strPDF As String

On Error Resume Next

Screen.MousePointer = vbHourglass

strPDF = AppPath & "\multiple.pdf"

Kill strPDF
Err.Clear 'clear "file not found" error

'Create a single PDF from all PDFs files in directory matching *.pdf
oPDFmaker.CreatePDFfromMultiPDFinDir strPDF, AppPath, "*.pdf", pdoSortByName, pdoSortOrderAscending

Screen.MousePointer = vbDefault

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

Example:

```
Dim strPDF As String
Dim strPS(1) As String
Dim strPrinter As String
Dim strWB As String
Dim strRange As String
Dim strDoc As String
Dim strPagesToPrint As String
```

```
On Error Resume Next
```

```
Screen.MousePointer = vbHourglass
```

```
'PDF file to create
strPDF = App.Path & "\multiple.pdf"
```

```
'PostScript files to create
strPS(0) = App.Path & "\sample.ps"
strPS(1) = App.Path & "\debug.ps"
```

```
'Printer driver name
strPrinter = "Generic Postscript Printer"
```

```
'Excel workbook
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
```

```
'Word document
strDoc = App.Path & "\debug.doc"
'Pages to use
strPagesToPrint = "1, 3, 6-8, 10"
```

```
'Create PostScript file from Excel worksheet
oPDFActiveX.CreatePSfromExcel strPS(0), strWB, 1, strRange, , , , , "Page &P", pdoAlignRight, "&D &T",
pdoAlignCenter, 1#, 0.5, 0#, 1#
```

```
'Create PostScript file from Word document
oPDFActiveX.CreatePSfromWord strPS(1), strDoc, , , , strPagesToPrint
```

```
'Combine PostScript files and create a PDF file
oPDFActiveX.CreatePDFfromMultiPS strPDF, strPS
```

```
Screen.MousePointer = vbDefault
```

```
If Err.Number Then
    MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
    'open PDF file
    ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1
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 strPrinter As String	Print driver. Default: "Generic PostScript Printer"
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 blnBlackAndWhite As Boolean	Black and white?

Example:

```
Dim strPDF As String
Dim strDoc As String
Dim strPrinter As String
```

On Error Resume Next

Screen.MousePointer = vbHourglass

```
strPDF = App.Path & "\file.pdf"
strDoc = App.Path & "\file.ppt"
strPrinter = "Generic Postscript Printer"
```

oPDFActiveX.CreatePDFfromPowerPoint strPDF, strDoc, strPrinter

If Err.Number Then

MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"

MsgBox "Powerpoint must be installed on this machine.", vbInformation, "Message"

End If

Screen.MousePointer = vbDefault

If Err.Number Then

MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"

Else

'open PDF file

ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1

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

Example:

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

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

```
oPDFActiveX.CreatePSfromExcel strPS, strWB, 1, strRange, , , , , "Page &P", pdoAlignRight, "&D &T",
pdoAlignCenter, 1#, 0.5, 0#, 1#
```

If Err.Number Then

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

'create a PDF file from a PostScript file

```
oPDFActiveX.CreatePDFfromPS strPDF, strPS
```

Screen.MousePointer = vbDefault

If Err.Number Then

```
MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
Else
'open PDF file
ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1
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 strPrinter As String	Print driver. Default: "Generic PostScript Printer"
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

Example:

```
Dim strPDF As String
Dim strTXT As String
Dim strPrinter As String
```

On Error Resume Next

```
Screen.MousePointer = vbHourglass
```

```
strPDF = App.Path & "\file.pdf"
strTXT = App.Path & "\file.txt"
strPrinter = "Generic Postscript Printer"
```

'CreatePDFfromTextFile requires Microsoft Word

'use methods CreatePDFfromTextFile2 or CreatePDFfromTextFile3

'if you don't have Microsoft Word

```
oPDFActiveX.CreatePDFfromTextFile strPDF, strTXT, strPrinter, False, "arial", False, 12, RGB(0, 0, 255), False,
False, False, True, pdoAlignRight, 20.5, 20.5, , 20.5
```

If Err.Number Then

```
MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
```

```
MsgBox "Word must be installed on this machine.", vbInformation, "Message"
```

End If

```
Screen.MousePointer = vbDefault
```

If Err.Number Then

```
MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
```

Else

```
'open PDF file
```

```
ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1
```

End If

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

Requirements: Notepad and file PDFSEND.EXE

Arguments:

strPDFFile As String	PDF file to create
strTextFile As String	Text file to use
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional strSaveWindowCaption As String	Window caption. Default: "print to file"

Example:

```
Dim strPDF As String
Dim strTXT As String
Dim strPrinter As String
```

```
On Error Resume Next
```

```
Screen.MousePointer = vbHourglass
```

```
strPDF = App.Path & "\file.pdf"
strTXT = App.Path & "\file.txt"
strPrinter = "Generic Postscript Printer"
```

```
'CreatePDFfromTextFile requires Microsoft Word
'use methods CreatePDFfromTextFile2 or CreatePDFfromTextFile3
'if you don't have Microsoft Word
oPDFActiveX.CreatePDFfromTextFile2 strPDF, strTXT
If Err.Number Then
    MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
    MsgBox "Notepad must be installed on this machine.", vbInformation, "Message"
End If
```

```
Screen.MousePointer = vbDefault
```

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

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

Requirements: File PDFSEND.EXE

Arguments:

strPDFFile As String	PDF file to create
----------------------	--------------------

strTextfile As String	Text file to use
Optional intRightMargin As Integer	Right margin
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional strSaveWindowCaption As String	Window caption. Default: "print to file"

Example:

```
Dim strPDF As String
Dim strTXT As String
Dim strPrinter As String
```

```
On Error Resume Next
```

```
Screen.MousePointer = vbHourglass
```

```
strPDF = App.Path & "\file.pdf"
strTXT = App.Path & "\file.txt"
strPrinter = "Generic Postscript Printer"
```

```
'CreatePDFfromTextFile requires Microsoft Word
'use methods CreatePDFfromTextFile2 or CreatePDFfromTextFile3
'if you don't have Microsoft Word
oPDFActiveX.CreatePDFfromTextFile3 strPDF, strTXT
If Err.Number Then
    MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
    MsgBox "Word must be installed on this machine.", vbInformation, "Message"
End If
```

```
Screen.MousePointer = vbDefault
```

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

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

Requirements: Visio and file PDFSEND.EXE

Arguments:

strPDFFile As String	PDF file to create
strVisioFile As String	Visio file to use
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional strSaveWindowCaption As String	Window caption. Default: "print to file"

Example:

```
Dim strPDF As String
```

```
Dim strVSD As String
Dim strPrinter As String
```

```
On Error Resume Next
```

```
Screen.MousePointer = vbHourglass
```

```
strPDF = App.Path & "\bflowcht.pdf"
strVSD = App.Path & "\bflowcht.vsd"
strPrinter = "Generic Postscript Printer"
```

```
oPDFActiveX.CreatePDFfromVisio strPDF, strVSD, strPrinter
```

```
Screen.MousePointer = vbDefault
```

```
If Err.Number Then
```

```
MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
```

```
MsgBox "Visio must be installed on this machine.", vbInformation, "Message"
```

```
Else
```

```
'open PDF file
```

```
ShellExecute 0, vbNullString, strPDF, vbNullString, vbNullString, 1
```

```
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 strPrinter As String	Print driver. Default: "Generic PostScript Printer"
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

Example:

```
Dim strPDF As String
Dim strDoc As String
Dim strPrinter As String
```

```
On Error Resume Next
```

```
Screen.MousePointer = vbHourglass
```

```
strPDF = App.Path & "\debug.pdf"  
strDoc = App.Path & "\debug.doc"  
strPrinter = "Generic Postscript Printer"
```

```
oPDFActiveX.CreatePDFfromWord strPDF, strDoc, strPrinter
```

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

```
Screen.MousePointer = vbDefault
```

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

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

Requirements: Access and file PDFOSEND.EXE

Arguments:

strPSFile As String	Postscript file to create
strDatabase As String	Access database to use
strReportNameAs String	Access report to use
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional strPassword As String	Access database password
Optional strFilter As String	Access report filter
Optional strWhereCondition As String	Access report where condition
Optional strSaveWindowCaption As String	Window caption. Default: "print to file"

Example:

```
Dim strPS As String  
Dim strMDB As String  
Dim strRPT As String  
Dim strPrinter As String  
On Error Resume Next
```

```
Screen.MousePointer = vbHourglass  
strPS = App.Path & "\Report.ps"  
strMDB = App.Path & "\Author.mdb"  
strPrinter = "Generic Postscript Printer"
```

'Note: You may need to modify this file path

```
'strMDB = "C:\Program Files\Microsoft Visual Studio\VB98\NWIND.MDB"
```

'If you are using a non-English version of Windows

'you can modify parameter strSaveWindowCaption

'Use code below to create a PDF from a single report

```
strRPT = "authors"  
oPDFActiveX.CreatePSfromAccessReport strPS, strMDB, strRPT, strPrinter  
Screen.MousePointer = vbDefault  
If Err.Number Then  
    MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"  
    MsgBox "Access must be installed on this machine.", vbInformation, "Message"  
End If
```

CreatePSfromAutoCAD() - Method to create a PostScript (PS) file from an AutoCAD drawing.

Requirements: Access and file PDFSEND.EXE

Arguments:

strPSFile As String	Postscript file to create
strDWGFile As String	Drawing file to use
Optional varLayout As Variant	Layout to use
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional strSaveWindowCaption As String	Window caption. Default: "print to file"

Example:

```
Dim strPS As String  
Dim strDWG As String  
Dim i As Integer  
Dim strPassword As String  
Dim strPrinter As String
```

On Error Resume Next

```
strPS = App.Path & "\HelloWorld.ps"  
strDWG = App.Path & "\HelloWorld.dwg" 'can also use DXF files  
strPrinter = "Generic Postscript Printer"
```

```
oPDFActiveX.CreatePSfromAutoCAD strPS, strDWG, "layout1", strPrinter
```

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

CreatePSfromCrystalReport() - Method to create a PostScript (PS) file from a Crystal Report.

Requirements: Crystal Reports and file PDFSEND.EXE

Arguments:

strPSFile As String	Postscript file to create
strCrystalReport As String	Report to use
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional strSaveWindowCaption As String	Window caption. Default: "print to file"

Example:

```
Dim strRPT As String
Dim strPS As String
Dim i As Long
Dim strPrinter As String
```

On Error Resume Next

```
strRPT = App.Path & "\sample.rpt"
strPS = App.Path & "\sample.ps"
strPrinter = "Generic Postscript Printer"
```

```
oPDFActiveX.CreatePSfromCrystalReport strPS, strRPT, strPrinter
```

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

```
If Err.Number Then
    MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
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 strPrinter As String	Print driver. Default: "Generic PostScript Printer"
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

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"
```

"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
```

```
oPDFActiveX.CreatePSfromExcel strPS, strWB, 1, strRange, , , , , "Page &P", pdoAlignRight, "&D &T",
pdoAlignCenter, 1#, 0.5, 0#, 1#
```

```
Screen.MousePointer = vbDefault
```

```
If Err.Number Then
```

```
MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
```

```
MsgBox "Excel must be installed on this machine.", vbInformation, "Message"
```

```
Else
```

```
MsgBox "PostScript file created.", vbInformation, "Message"
```

```
End If
```

CreatePSfromImageFile() - Method to create a PostScript (PS) file from an image file.

Requirements: File PDFSEND.EXE

Arguments:

strPSFile As String	Postscript file to create
strImagefile As String	Image file to use
Optional sngXpos As Single	X position
Optional sngYpos As Single	Y position
Optional sngWidth As Single	Width in pixels
Optional sngHeight As Single	Height in pixels
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional strSaveWindowCaption As String	Window caption. Default: "print to file"

Example:

```
Dim strPS As String
Dim strJPG As String
Dim strPrinter As String
```

```
On Error Resume Next
```

```
Screen.MousePointer = vbHourglass
```

```
strPS = App.Path & "\file.ps"
strJPG = App.Path & "\image1.jpg"
strPrinter = "Generic Postscript Printer"
```

```
oPDFActiveX.CreatePSfromImageFile strPS, strJPG, 100, 1000, 0, 0, strPrinter
```

```
If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
End If
```

```
Screen.MousePointer = vbDefault
```

```
If Err.Number Then
  MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error 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 strPrinter As String	Print driver. Default: "Generic PostScript Printer"
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 blnBlackAndWhite As Boolean	Black and white?

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"
```


oPDFActiveX.CreatePSfromPowerPoint strPS, strDoc

Screen.MousePointer = vbDefault

If Err.Number Then

MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"

MsgBox "Powerpoint must be installed on this machine.", vbInformation, "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 strPrinter As String	Print driver. Default: "Generic PostScript Printer"
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

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"

oPDFActiveX.CreatePSfromTextFile strPS, strTXT, , False, "arial", False, 12, RGB(0, 0, 255), False, False, False, True, pdoAlignRight, 20.5, 20.5, , 20.5

```
Screen.MousePointer = vbDefault
```

```
If Err.Number Then
```

```
    MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
```

```
    MsgBox "Word must be installed on this machine.", vbInformation, "Message"
```

```
Else
```

```
    MsgBox "PostScript file created.", vbInformation, "Message"
```

```
End If
```

```
*****
```

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

Requirements: Notepad and PDFSEND.EXE

Arguments:

strPSFile As String	Postscript file to create
strTextFile As String	Text file to use
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional strSaveWindowCaption As String	Window caption. Default: "print to file"

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"
```

```
oPDFActiveX.CreatePSfromTextFile2 strPS, strTXT  
Screen.MousePointer = vbDefault
```

```
If Err.Number Then
```

```
    MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
```

```
    MsgBox "Notepad must be installed on this machine.", vbInformation, "Message"
```

```
Else
```

```
    MsgBox "PostScript file created.", vbInformation, "Message"
```

```
End If
```

```
*****
```

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

Requirements: File PDFSEND.EXE

Arguments:

strPSFile As String	Postscript file to create
strTextFile As String	Text file to use
Optional intRightMargin As Integer	Right margin.
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional strSaveWindowCaption As String	Window caption. Default: "print to file"

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"
```

```
oPDFActiveX.CreatePSfromTextFile3 strPS, strTXT
Screen.MousePointer = vbDefault
```

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

CreatePSfromVisio() - Method to create a PostScript (PS) file from a Visio document.

Requirements: File PDFSEND.EXE

Arguments:

strPSFile As String	Postscript file to create
strVisioFile As String	Visio document to use
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional strSaveWindowCaption As String	Window caption. Default: "print to file"

Example:

```
Dim strPS As String
Dim strVSD As String
Dim strPrinter As String
```

```
On Error Resume Next
```

```
Screen.MousePointer = vbHourglass
```

```
strPS = App.Path & "\bflowcht.ps"
strVSD = App.Path & "\bflowcht.vsd"
strPrinter = "Generic Postscript Printer"
```

```
oPDFActiveX.CreatePSfromVisio strPS, strVSD, strPrinter
```

```
Screen.MousePointer = vbDefault
```

```
If Err.Number Then
```

```
    MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
```

```
    MsgBox "Visio must be installed on this machine.", 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 strPrinter As String	Print driver. Default: "Generic PostScript Printer"
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

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"
```

```
oPDFActiveX.CreatePSfromWord strPS, strDoc, , , strPagesToPrint
```

```
Screen.MousePointer = vbDefault
```

```
If Err.Number Then
```

```
    MsgBox Err.Number & " : " & Err.Description, vbExclamation, "Error Message"
```

```
    MsgBox "Word must be installed on this machine.", vbInformation, "Message"
```

```
Else
```

```
    MsgBox "PostScript file created.", vbInformation, "Message"
```

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
oPDFActiveX.DownloadFile Inet1, "microsoft.com", "/mydirectory", "c:\file.txt", "file.txt"

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

Arguments:

strPrinter As String Printer to use

Example:

oPDFActiveX.SetDefPrinter "HP DeskJet 890C"

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
oPDFActiveX.UploadFile Inet1, "www.microsoft.com", "/mydirectory", App.Path & "\table.htm", "table.htm",
"username", "password"

There is version of PDF ActiveX DLL available for web developers which will run on a web server named PDF ActiveX DLL for ASP. You can download it from here:

<http://www.getfilez.com/pdfxasp.zip>

There is a .NET version of PDF ActiveX DLL named PDF Create .NET for Visual Basic .NET and C# developers available at:

<http://www.getfilez.com/setup.msi>

SkySof Software Inc.

<http://www.skysof.com>

Email: kusluski@bellsouth.net