

PDF Create .NET

Author: SkySof Software

Last Updated: 7/11/2007

Before using PDF Create .NET 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 Create .NET is that you can develop applications to produce PDF files which don't require Adobe Acrobat or any other expensive third party software which must be purchased. Also, you can install your application on as many machines as you like without paying any royalty fees! PDF Create .NET is a great, low-cost software development solution for large companies that need PDF publishing capabilities. You'll be amazed how quickly this DLL produces high quality PDF files!

- 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>

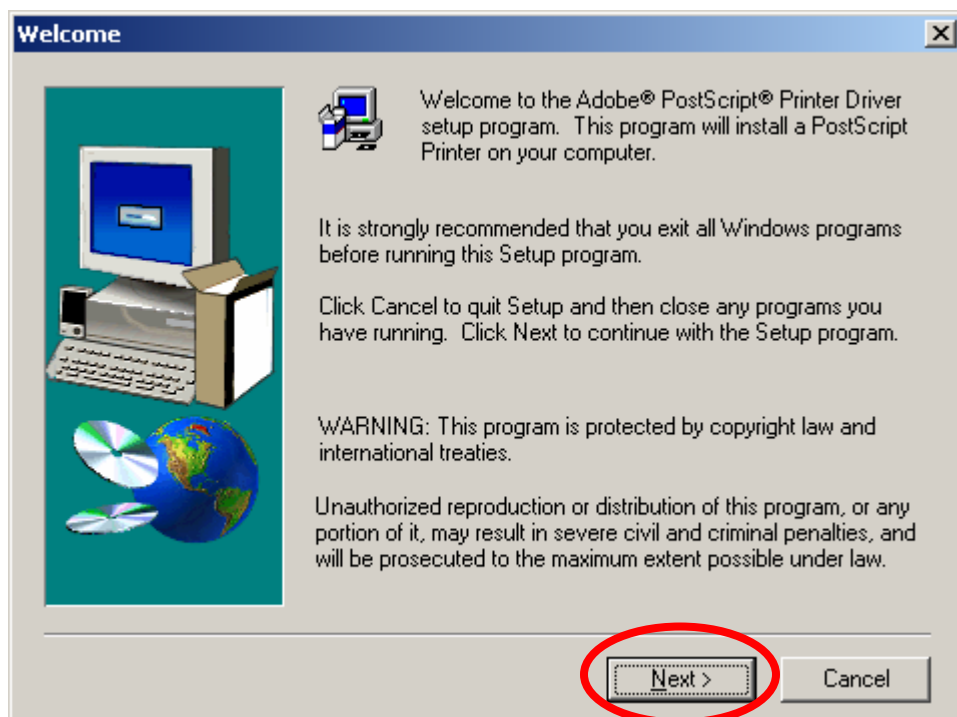
(look for **PostScript printer drivers**)

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

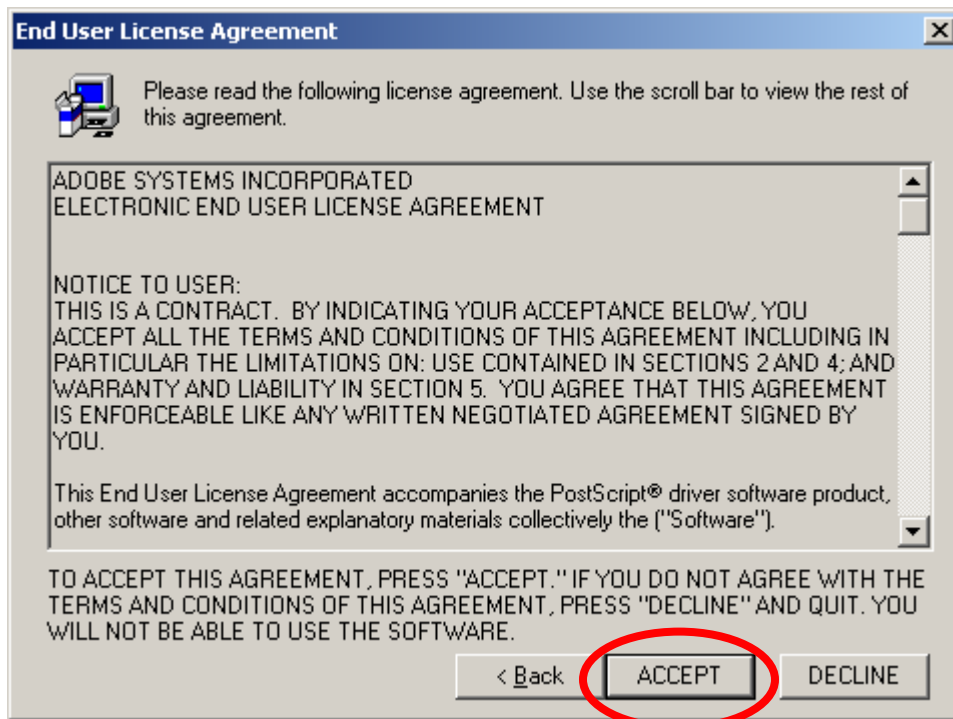
<http://www.getfilez.com/winsteng.exe>

To install the PostScript Print Driver:

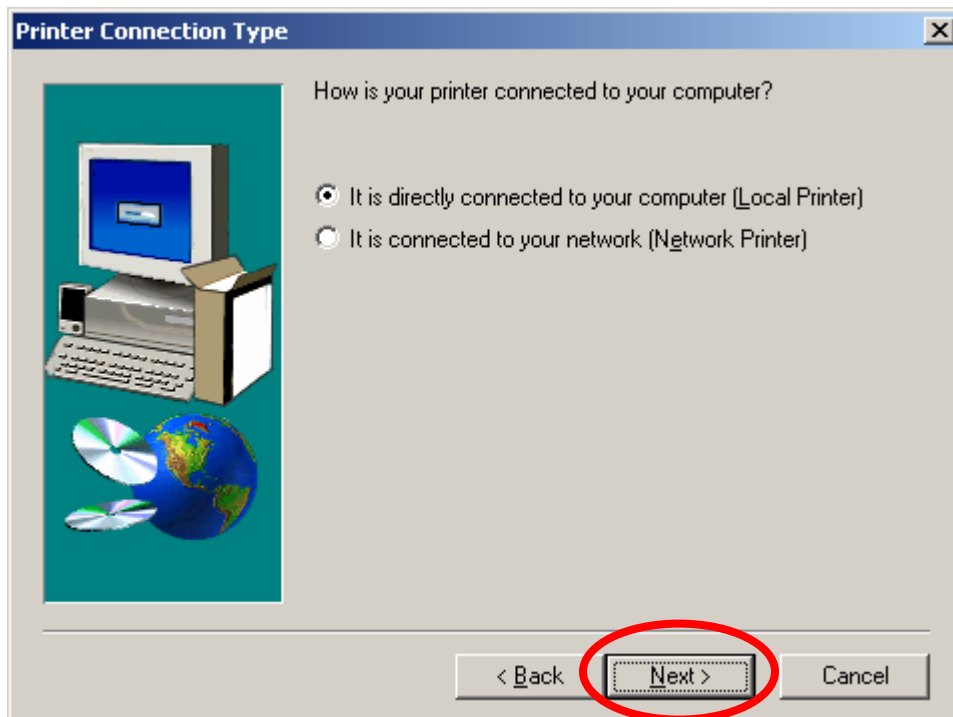
- a) Run the installation file `winsteng.exe`
- b) The following screen will appear. Click the **Next** Button.



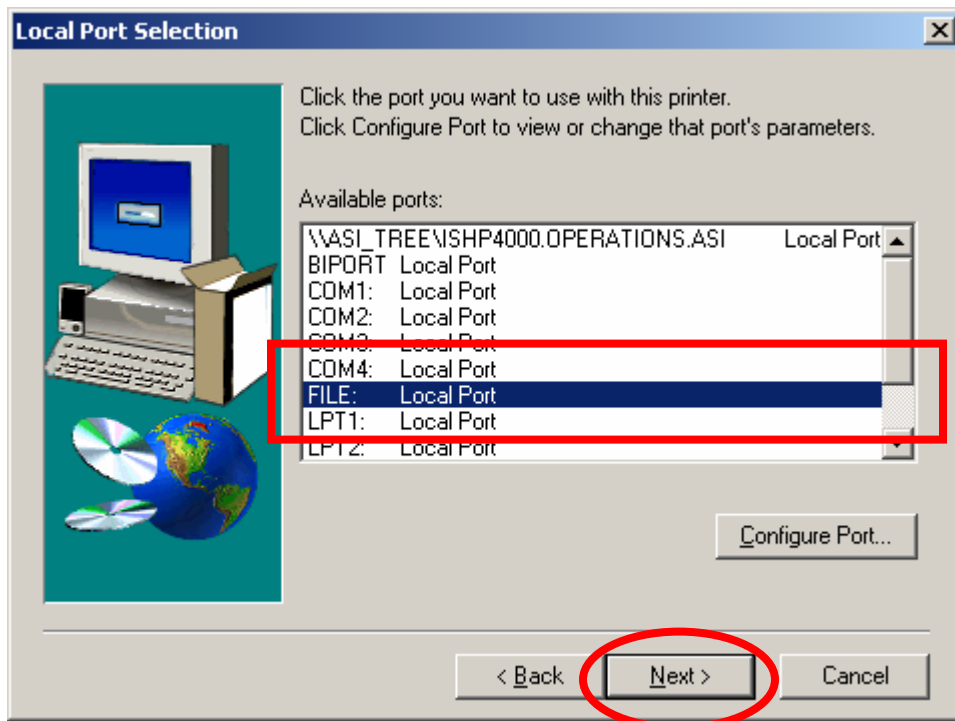
c) Click the **Accept** Button



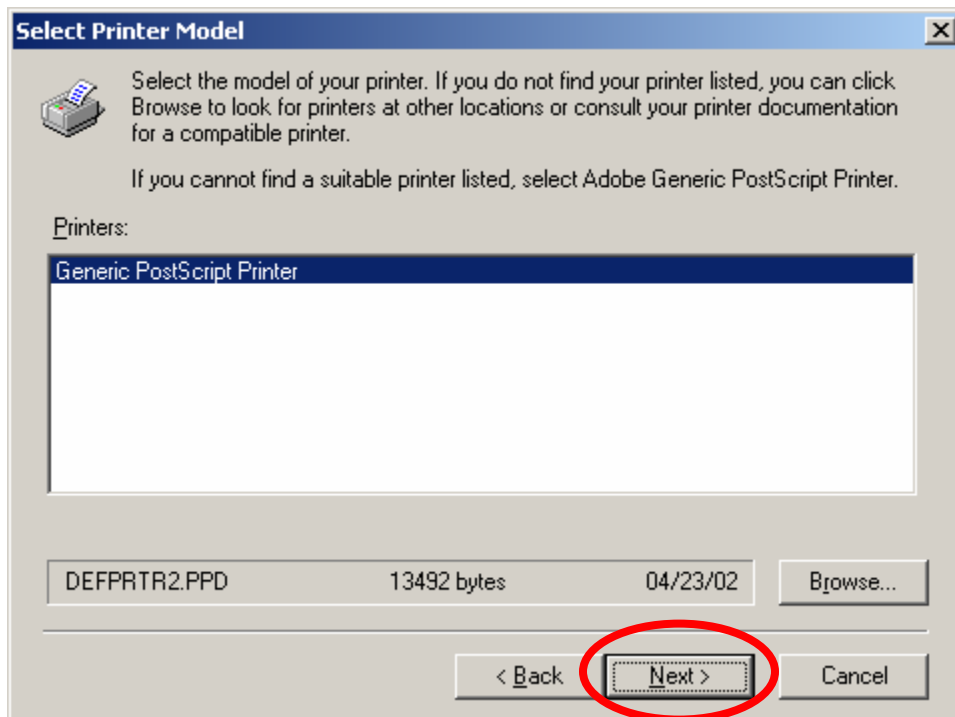
d) Click the **Next** Button



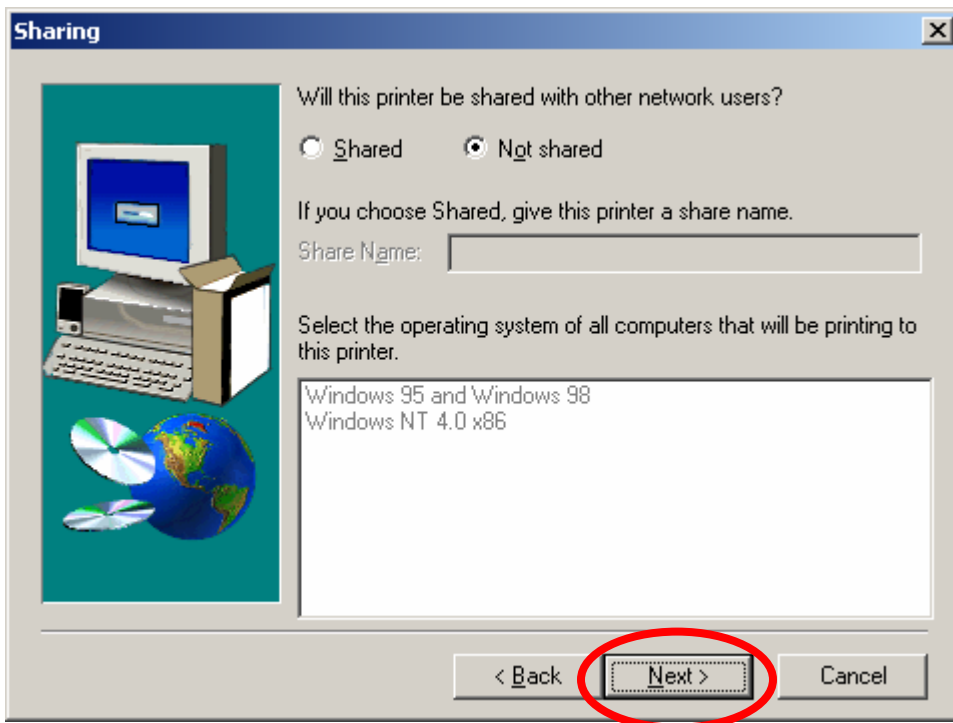
e) Select **FILE:** and click the **Next** Button



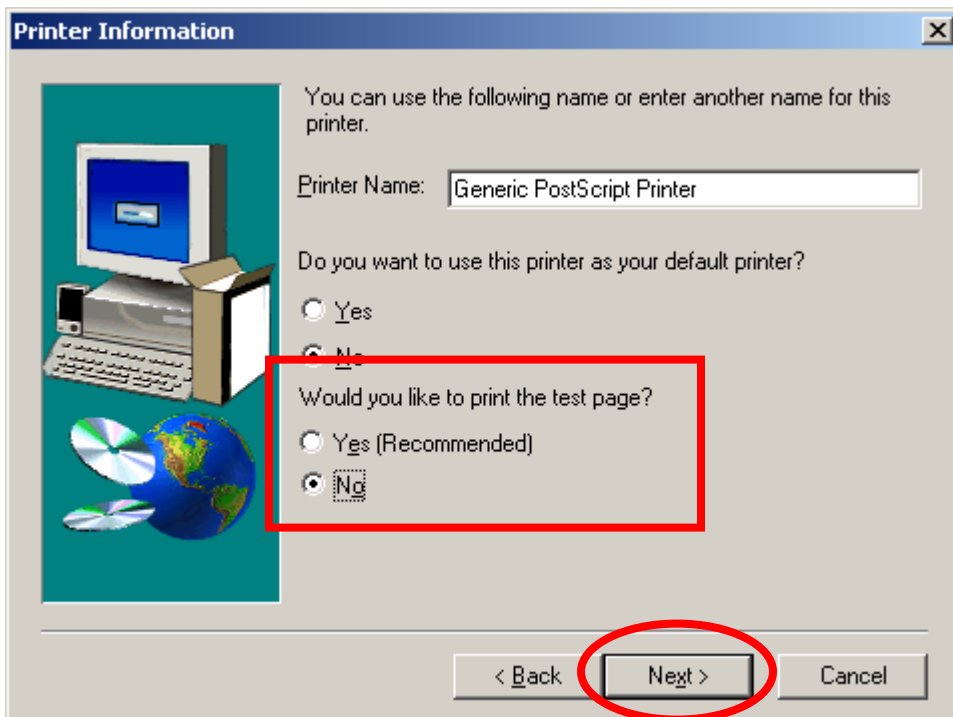
f) Click the **Next** Button



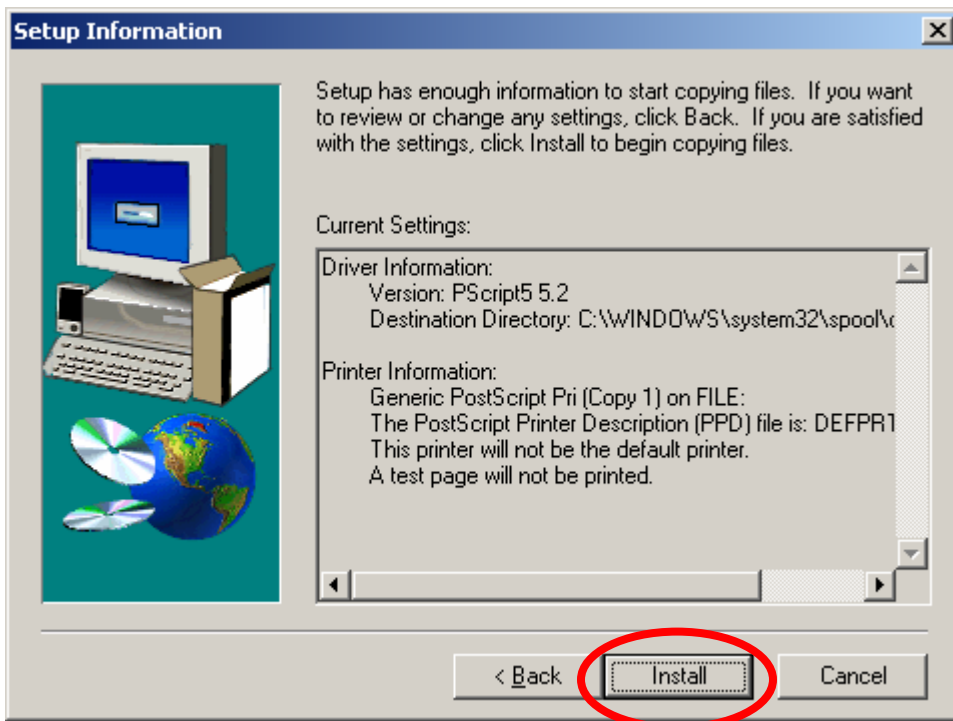
g) Click the **Next** Button



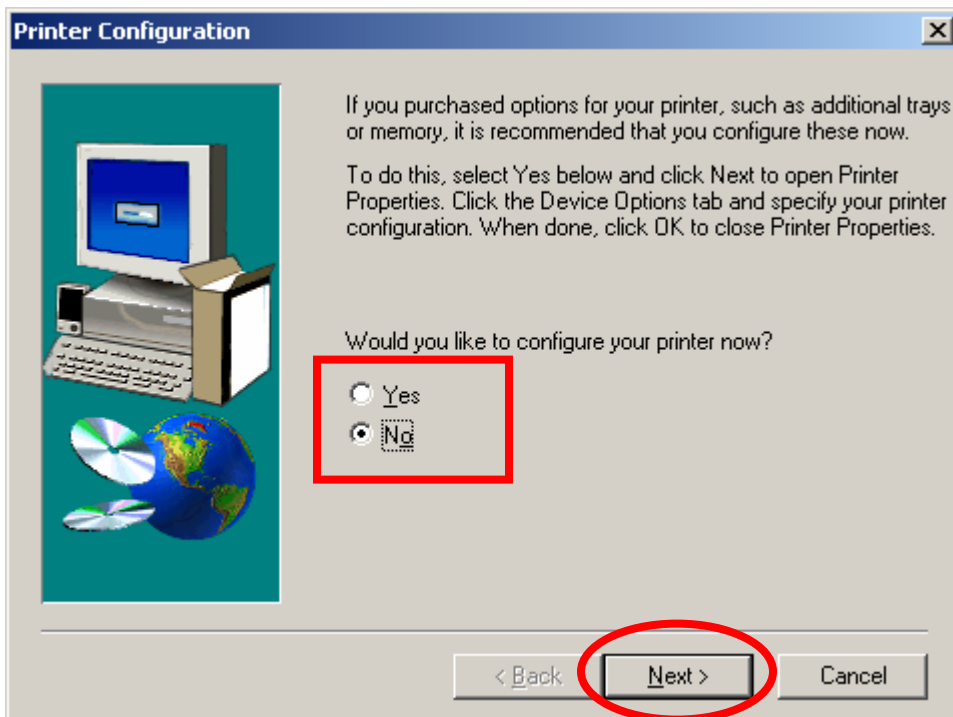
h) Select **No** and click the **Next** Button



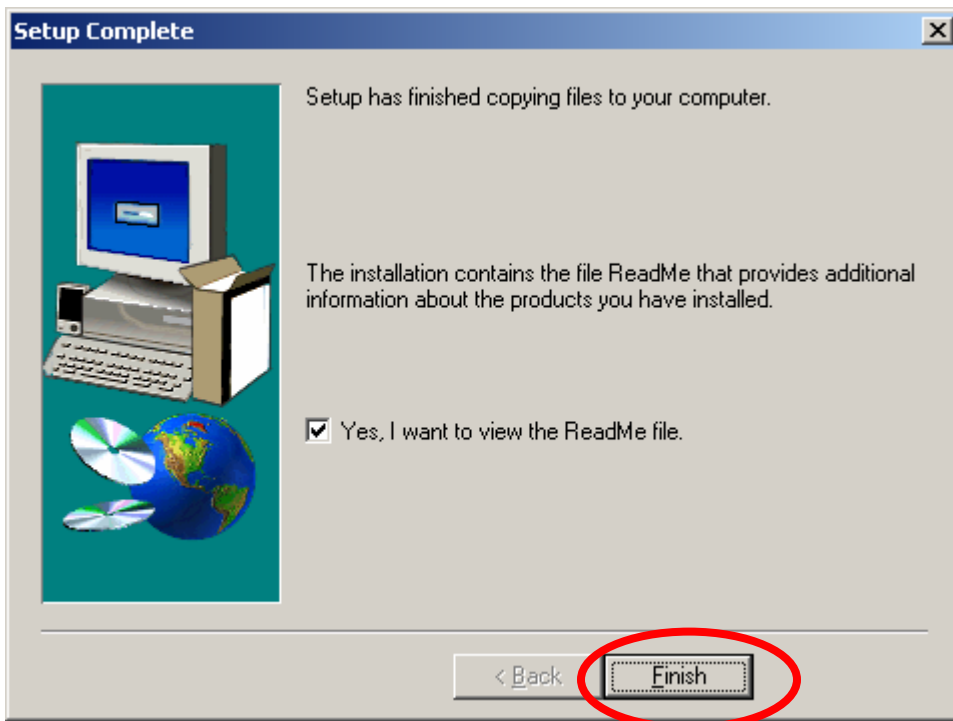
i) Click the **Install** Button



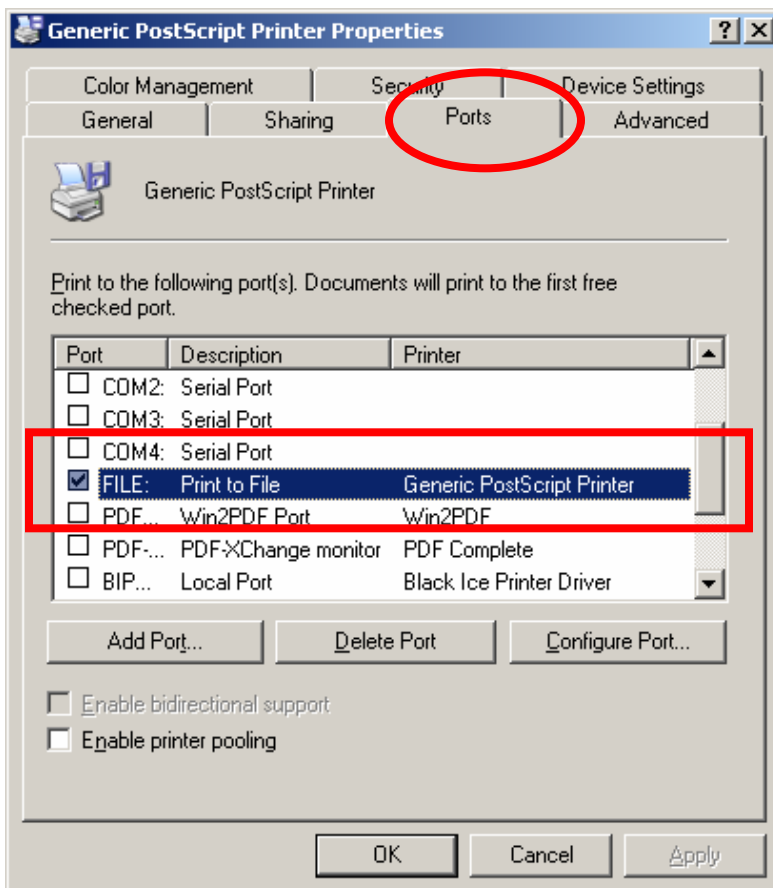
j) Select **No** and click the **Next** Button



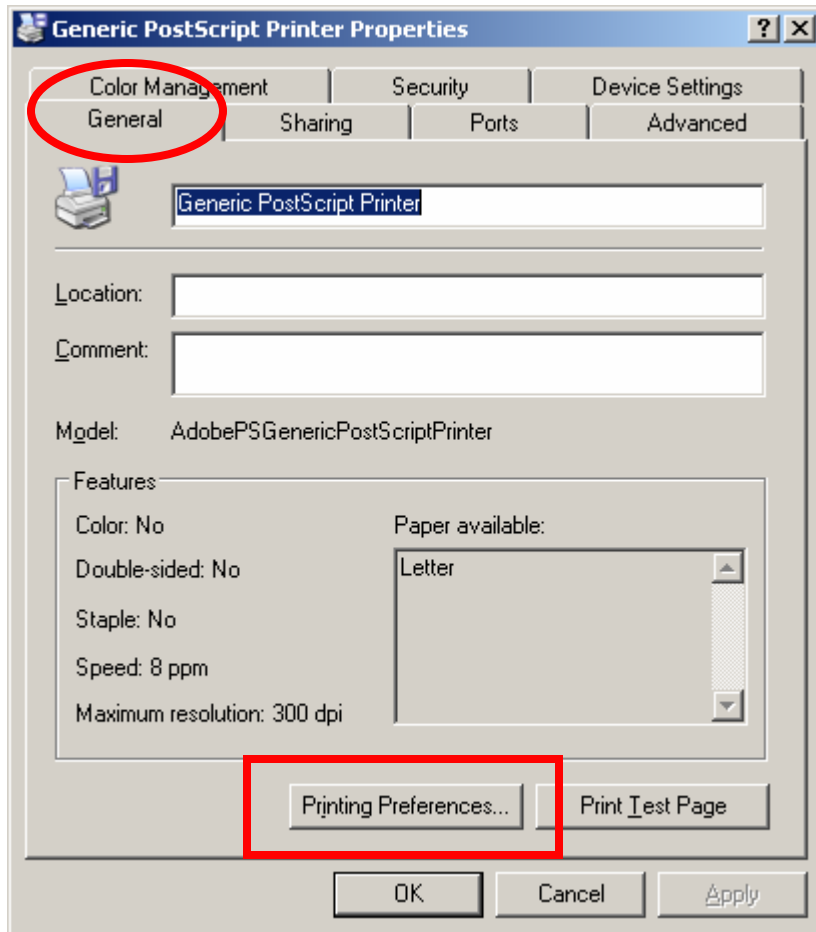
k) Click the **Finish** Button



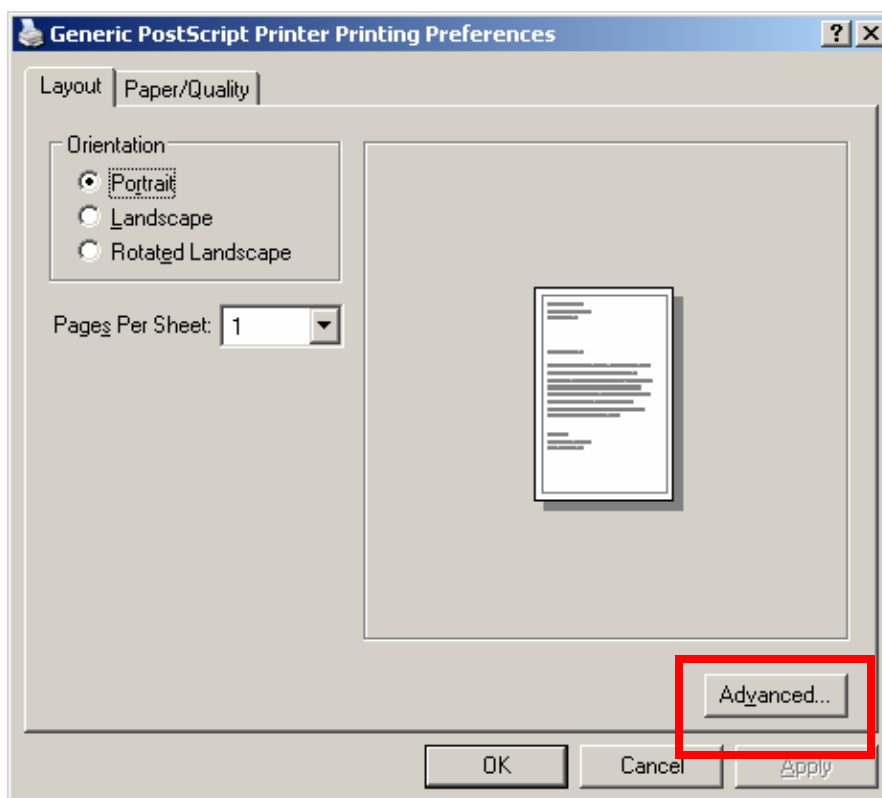
2) After installing the printer driver you will need to modify settings for the PostScript Print Driver. To do so click Window's Start Button, select **Printers and Faxes** from the menu, right-click on **Generic PostScript Printer** and select **Properties** from the popup menu, then click the Ports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



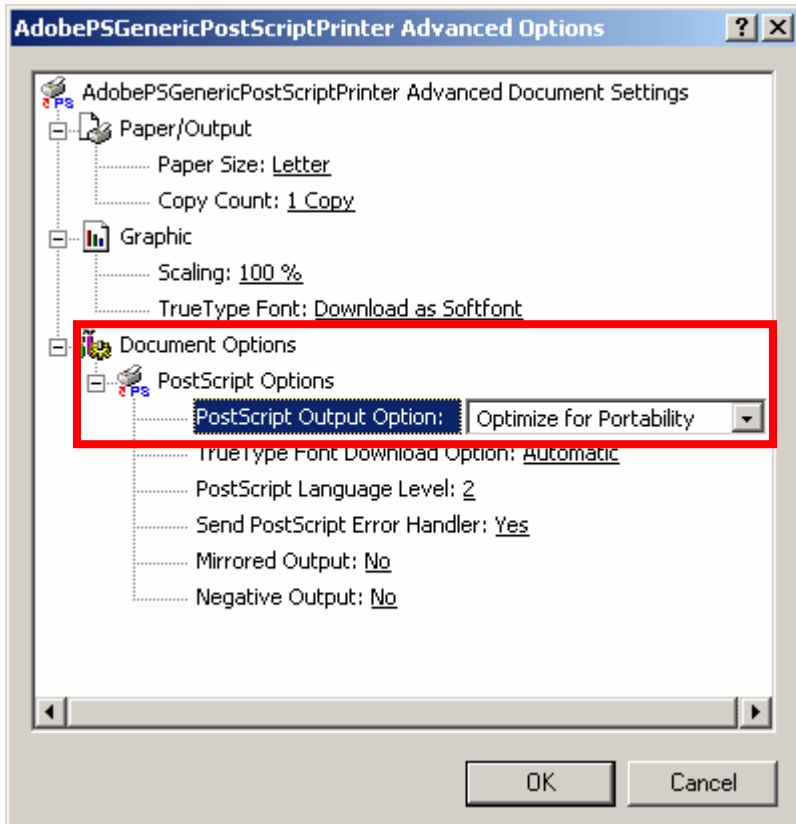
- 3) If you want your PDF files to appear in color you will need to make another change. To do so click the **General Tab** and click the **Printing Preferences** Button.



- 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>

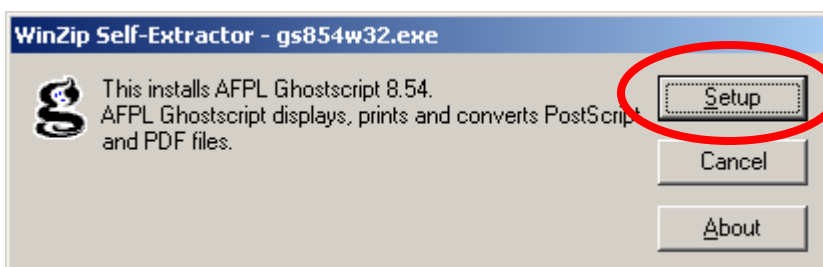
We recommend that you install GhostScript version 8.54 which is the most recent version at the time of this writing. You can download the 32-bit Window's version of GhostScript 8.54 by clicking on this link

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

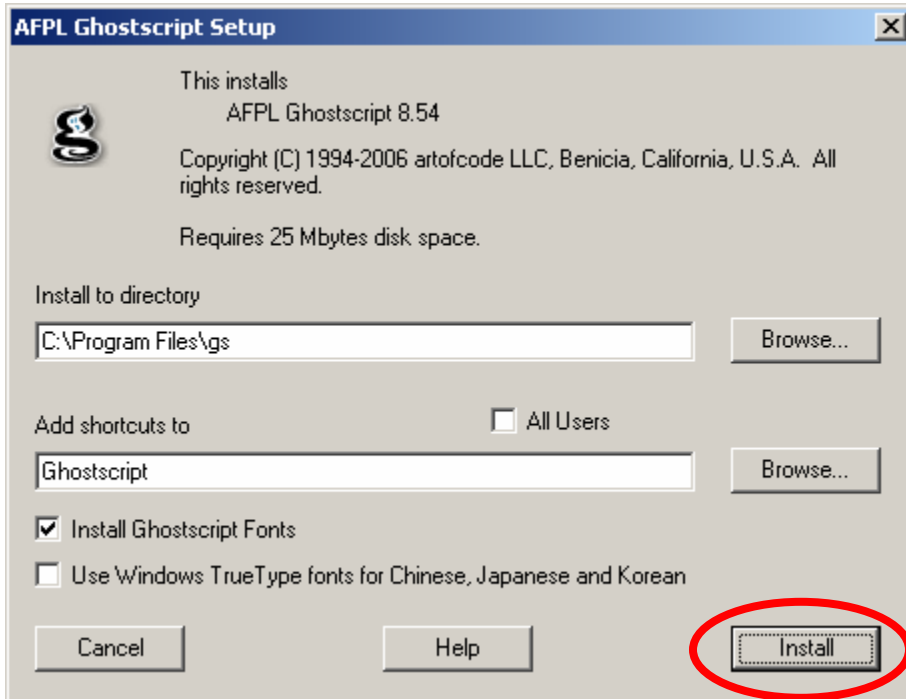
<http://www.getfilez.com/g854w64.exe>

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 Create .NET in your .NET Projects

- 1) Load .NET and create a new Window's Application (Visual Basic or C#)
- 2) From the Project Menu select Add Reference...
- 3) Click the Browse Button
- 4) Select C:\Program Files\SkySof Software Inc\PDF Create .NET\pdfcnet.dll
- 5) Click the OK Button
- 6) At the top of the form type:

```
Imports SkySof          ' Visual Basic
using SkySof;          // C#
```

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

```
Dim oPDF As PDFCreate    ' Visual Basic
PDFCreate oPDF;         // C#
```

- 8) In the Form1_Load subroutine enter code:

```
oPDF = New PDFCreate    ' Visual Basic
oPDF.TempFilepath = "C:\Temp"
oPDF = new PDFCreate(); // C#
oPDF.TempFilePath = @"C:\Temp";
```

Distributing PDF Create .NET with your application

PDF Create .NET may be distributed with your application royalty free. Some functions which access the PostScript print driver require file PDFSEND.EXE in your setup program. This file is located in your Window's System32 Directory (typically C:\Windows\System32). You should install this file on the client's machine in the

same directory as the PDF Create .NET DLL or in their Window's System32 Directory. The file PDFSEND.EXE is used by PDF Create .NET to send messages to the "Print to File" window which pops up when printing with the PostScript print driver.

Note: All subsequent code in this document is Visual Basic .NET

Properties

GhostScriptVersion – Get GhostScript version number (String).

PDFPassword – Get/set PDF password (String). Used for creating secure PDF files.

PDFPermissions – Get/set PDF permission (Integer). Used for setting PDF file permissions.

PDFVersion – Get/set PDF version (Integer). Values 0 – 3 used for creating specific PDF file type.

PrinterName – Get default printer name (String).

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

Events

DoneCreatingPDF – Fired after PDF file has been created.

DoneCreatingPS – Fired after PS file has been created.

StartCreatingPDF – Fired before creating PDF file.

StartCreatingPS – Fired before creating PS file.

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 = AppPath & "\Report.pdf"
Dim strMDB As String = AppPath & "\Author.mdb"
Dim strRPT As String = "authors"

Try

    oPDF.CreatePDFfromAccessReport(strPDF, strMDB, strRPT)
    oPDF.GarbageCollect()
    System.Diagnostics.Process.Start(strPDF)

Catch ex As Exception

    MsgBox(ex.Message)

End Try
```

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 objLayout As Object	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 = AppPath & "\HelloWorld.pdf"
Dim strDWG As String = AppPath & "\HelloWorld.dwg" 'can also use DXF files

Try

    oPDF.CreatePDFfromAutoCAD(strPDF, strDWG, "layout1")
    oPDF.GarbageCollect()
    System.Diagnostics.Process.Start(strPDF)

Catch ex As Exception

    MsgBox(ex.Message)

End Try
```

CreatePDFfromCrystalReport() - Method to create an Adobe Acrobat (PDF) file from a Crystal Report File. Note: this method uses the Crystal Report Engine packaged with Visual Studio .NET to generate a PDF.

Requirements: Crystal Reports

Arguments:

strPDFFile As String	PDF file to create
strCrystalReport As String	Crystal Report File to use
Optional strRecSelFormula As String	Record selection formula
Optional strUsername As String	Database login username
Optional strPassword As String	Database login password

Example:

```
Dim strRPT As String = AppPath & "\chart.rpt"
Dim strPDF As String = AppPath & "\sample.pdf"

Try

    oPDF.CreatePDFfromCrystalReport(strPDF, strRPT)
    System.Diagnostics.Process.Start(strPDF)
```

```

Catch ex As Exception
    MsgBox (ex.Message)

End Try

```

CreatePDFfromCrystalReport2() - Method to create an Adobe Acrobat (PDF) file from a Crystal Report File. This method uses COM.

Requirements: Crystal Reports and file PDFSEND.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 = AppPath & "\sample.rpt"
Dim strPDF As String = AppPath & "\sample.pdf"

Try

    oPDF.CreatePDFfromCrystalReport2(strPDF, strRPT)
    oPDF.GarbageCollect()
    System.Diagnostics.Process.Start(strPDF)

Catch ex As Exception

    MsgBox (ex.Message)

End Try

```

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
objExcelWorksheet As Object	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 Integer	Page header alignment. Values: 0=left, 1=center, 2=right

Optional strFooter As String	Page footer
Optional lngFooterAlignment As Integer	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 intFirstPageNumber As Integer	Beginning page number

Example:

```

Dim strPDF As String = AppPath & "\sample.pdf"
Dim strWB As String = AppPath & "\sample.xls"
Dim strRange As String
'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

Try

    oPDF.CreatePDFfromExcel(strPDF, strWB, 1, strRange, "", "", "", False,
False, "Page &P", PDFCreate.AlignType.pdoAlignRight, "&D &T",
PDFCreate.AlignType.pdoAlignCenter, 1.0, 0.5, 0.0, 1.0, 0.0, 0.0, 1)
    'oPDF.CreatePDFfromExcel(strPDF, strWB, "xtreme", strRange)
    oPDF.GarbageCollect()
    System.Diagnostics.Process.Start(strPDF)

Catch ex As Exception

    MsgBox(ex.Message)

End Try

```

CreatePDFfromImageFile() - Method to create an Adobe Acrobat (PDF) file from an image file (JPG,GIF,BMP,TIF,PNG,PCX,etc.). If arguments intX or intY are omitted the image will be resized to fit the page.

Requirements: File PDFSEND.EXE

Arguments:

strPDFFile As String	PDF file to create
strImageFile As String	Image file to use
Optional intX As Integer	X position
Optional intY As Integer	Y position
Optional intWidth As Integer	Width in pixels
Optional intHeight As Integer	Height in pixels
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional blnLandscape As Boolean	Use landscape mode?
Optional strSaveWindowCaption As String	Window caption. Default: "print to file"

Example:

```

Dim strPDF As String = AppPath & "\form.pdf"

```

```

Dim strImage As String = AppPath & "\form.tif"
'Show image at original size/format at page coordinates X = 0 and Y = 0
oPDF.CreatePDFfromImageFile(strPDF, strImage, "", False, 0, 0)
'Resize image to fit the entire page
'oPDF.CreatePDFfromImageFile(strPDF, strImage)
System.Diagnostics.Process.Start(strPDF)

```

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. Note: This function uses Window's file RNDLL32.EXE to send the image to the printer.

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 = AppPath & "\form.pdf"
Dim strImage As String = AppPath & "\form.tif"
'Resize image to fit the entire page
oPDF.CreatePDFfromImageFile2(strPDF, strImage)
System.Diagnostics.Process.Start(strPDF)

```

CreatePDFfromMiscFile() – Method to create a PDF file from various file types including TXT, RTF, HTML, etc.

Requirements: Word

Arguments:

strPDFFile As String	PDF file to create
strMiscFile As String	File to use
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"

Example:

```

Dim strPDF As String = AppPath & "\form.pdf"
Dim strRTF As String = AppPath & "\form.rtf"

Try

    oPDF.CreatePDFfromMiscFile(strPDF, strRTF)
    oPDF.GarbageCollect()
    System.Diagnostics.Process.Start(strPDF)

Catch ex As Exception

    MsgBox(ex.Message)

End Try

```

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

Requirements: Access and file PDFSEND.EXE

Note: Reports must exist in the same database

Arguments:

strPDFFile As String	PDF file to create
strDatabase As String	Database to use
arrReportNames As Array	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 = AppPath & "\Report.pdf"
Dim strMDB As String = AppPath & "\Author.mdb"
Dim arr(1) As String
arr(0) = "Authors"
arr(1) = "Sales"

Try

    oPDF.CreatePDFfromMultiAccessReport(strPDF, strMDB, arr)
    System.Diagnostics.Process.Start(strPDF)
    oPDF.GarbageCollect()

Catch ex As Exception

    MsgBox(ex.Message)

End Try
```

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
arrLayouts As Array	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 strPDF As String = AppPath & "\HelloWorld.pdf"
```

```

Dim strDWG As String = AppPath & "\HelloWorld.dwg" 'can also use DXF files
'Dim arr(2) As String
'arr(0) = "model"
'arr(1) = "layout1"
'arr(2) = "layout2"
Dim arr(2) As Integer
arr(0) = 0
arr(1) = 1
arr(2) = 2

```

Try

```

oPDF.CreatePDFfromMultiAutoCAD(strPDF, strDWG, arr)
oPDF.GarbageCollect()
System.Diagnostics.Process.Start(strPDF)

```

Catch ex As Exception

```

MsgBox(ex.Message)

```

End Try

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
arrExcelWorksheets As Array	Excel worksheets to use (array). Can be a name or number
arrRanges As Array	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 arrLandscape As Array	Page orientation (array). Values: True,False
Optional strHeader As String	Page header
Optional lngHeaderAlignment As Integer	Page header alignment. Values: 0=left, 1=center, 2=right
Optional strFooter As String	Page footer
Optional lngFooterAlignment As Integer	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 arrFirstPageNumber As Array	Beginning page numbers (array)

Example:

```

Dim strPDF As String
Dim strWB As String

```



```

Dim varWorksheets(2) As Short
Dim varRanges(2) As String
Dim varLandscape(2) As Boolean
Dim varFirstPageNumber(2) As Integer
Dim strPassword As String

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

'There are three ways to specify worksheet ranges

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

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

oPDF.CreatePDFfromMultiExcel(strPDF, strWB, varWorksheets, varRanges, "", "",
"", False, varLandscape, "Page &P", PDFCreate.AlignType.pdoAlignRight, "&D &T",
PDFCreate.AlignType.pdoAlignCenter, 1.0, 0.5, 0.0, 1.0, 0.0, 0.0, varFirstPageNumber)

oPDF.GarbageCollect()

System.Diagnostics.Process.Start(strPDF)

*****

```

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

Arguments:

strPDFFile As String	PDF file to create
arrPDFFiles As Array	PDF files to use (stored in one dimensional array)

Example:

```

Dim strPDF As String
Dim strPDFS(1) As String

'PDF file to create
strPDF = AppPath & "\multiple.pdf"

'PDF files to create
strPDFS(0) = AppPath & "\file1.pdf"
strPDFS(1) = AppPath & "\file2.pdf"

```

```

oPDF.CreatePDFfromWord(strPDFS(0), AppPath & "\debug.doc")
oPDF.CreatePDFfromExcel(strPDFS(1), AppPath & "\sample.xls", 1, "October", "",
"", "", False, False, "Page &P", PDFCreate.AlignType.pdoAlignRight, "&D &T",
PDFCreate.AlignType.pdoAlignCenter, 1.0, 0.5, 0.0, 1.0, 0.0, 0.0, 1)

```

'Combine multiple PDF files to create a single PDF file

```
oPDF.CreatePDFfromMultiPDF(strPDF, strPDFS)
```

```
oPDF.GarbageCollect()
```

```
System.Diagnostics.Process.Start(strPDF)
```

CreatePDFfromMultiPDFinDir() – Method to create a PDF file from multiple PDF files in a directory.

Arguments:

strPDFFile As String	PDF file to create
strDirectory As String	Directory to scan
strFiles as String	Files to scan for (can use wildcard characters * and ?)
lngSortType As Integer	Values: 0 = No sort, 1= Sort by filename

Example:

```

Dim strPDF As String = AppPath & "\manyindir.pdf"

Try
    'Create a single PDF from all PDFs files in directory matching *.pdf
    oPDF.CreatePDFfromMultiPDFinDir(strPDF, AppPath, "*.pdf",
PDFCreate.SortType.pdoSortByName)
    System.Diagnostics.Process.Start(strPDF)

Catch ex As Exception

    MsgBox(ex.Message)

End Try

```

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

Arguments:

strPDFFile As String	PDF file to create
arrPSFiles As Array	Postscript files to use (stored in one dimensional array)

Example:

```

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

'PDF file to create
strPDF = AppPath & "\multiple.pdf"

```

```

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

'Excel workbook
strWB = AppPath & "\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 = AppPath & "\debug.doc"
'Pages to use
strPagesToPrint = "1, 3, 6-8, 10"

'Create PostScript file from Excel worksheet
oPDF.CreatePSfromExcel(strPS(0), strWB, 1, strRange, "", "", "", False, False,
"Page &P", PDFCreate.AlignType.pdoAlignRight, "&D &T",
PDFCreate.AlignType.pdoAlignCenter, 1.0, 0.5, 0.0, 1.0, 0.0, 0.0, 1)

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

'Combine PostScript files and create a PDF file
oPDF.CreatePDFfromMultiPS(strPDF, strPS)

oPDF.GarbageCollect()
System.Diagnostics.Process.Start(strPDF)

```

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

Requirements: Power Point

Arguments:

strPDFFile As String	PDF file to create
strPPDocument As String	PowerPoint file to use
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional intStartSlide As Integer = -1	PowerPoint start slide to use. Use -1 for first slide
Optional intEndSlide As Integer = -1	PowerPoint end slide to use. Use -1 for last slide
Optional blnHorizontalOrientation As Boolean	PowerPoint slide orientation. Values: True,False
Optional blnBlackAndWhite As Boolean	Black and white?

Example:

```

Dim strPDF As String = AppPath & "\file.pdf"
Dim strDOC As String = AppPath & "\file.ppt"

Try

    oPDF.CreatePDFfromPowerPoint(strPDF, strDoc)
    oPDF.GarbageCollect()
    System.Diagnostics.Process.Start(strPDF)

Catch ex As Exception

    MsgBox(ex.Message)

```

End Try

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

strPS = AppPath & "\sample.ps"
strPDF = AppPath & "\sample.pdf"
strWB = AppPath & "\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
oPDF.CreatePSfromExcel(strPS, strWB, 1, strRange, "", "", "", False, False,
"Page &P", PDFCreate.AlignType.pdoAlignRight, "&D &T",
PDFCreate.AlignType.pdoAlignCenter, 1.0, 0.5, 0.0, 1.0, 0.0, 0.0, 1)

'create a PDF file from a PostScript file
oPDF.CreatePDFfromPS(strPDF, strPS)

oPDF.GarbageCollect()

System.Diagnostics.Process.Start(strPDF)
```

CreatePDFfromPub() – Method to create a PDF from a Publisher file.

Requirements: Publisher

Arguments:

strPDFFile As String	PDF file to create
strPubFile As String	Publisher file to use

Example:

```
Dim strPDF As String = AppPath & "\file.pdf"
Dim strPUB As String = AppPath & "\file.pub"

Try

oPDF.CreatePDFfromPub(strPDF, strPUB)
```

```
oPDF.GarbageCollect ()
System.Diagnostics.Process.Start (strPDF)
```

```
Catch ex As Exception
```

```
MsgBox (ex.Message)
```

```
End Try
```

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
Optional blnFontBold As Boolean	Font bold. Values: True,False
Optional intFontSize As Integer = 8	Font size
Optional lngFontColor As Integer	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 Integer	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 = AppPath & "\file.pdf"
Dim strTXT As String = AppPath & "\file.txt"

Try

    'CreatePDFfromTextFile requires Microsoft Word
    'use method CreatePDFfromTextFile2 if you don't have Microsoft Word
    oPDF.CreatePDFfromTextFile(strPDF, strTXT, "", False, "arial", False, 10,
    RGB(0, 0, 255), False, False, True, False, PDFCreate.AlignType.pdoAlignRight, 0.0,
    0.0, 0.0, 0.0, False)

    oPDF.GarbageCollect ()
    System.Diagnostics.Process.Start (strPDF)

Catch ex As Exception

    MsgBox (ex.Message)

End Try
```

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 = AppPath & "\file.pdf"
Dim strTXT As String = AppPath & "\file.txt"

Try

    'CreatePDFfromTextFile2 requires Notepad
    'also see method CreatePDFfromTextFile (requires Word)
    oPDF.CreatePDFfromTextFile2(strPDF, strTXT)
    System.Diagnostics.Process.Start(strPDF)

Catch ex As Exception

    MsgBox(ex.Message)

End Try
```

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 = AppPath & "\bflowcht.pdf"
Dim strVSD As String = AppPath & "\bflowcht.vsd"

Try

    oPDF.CreatePDFfromVisio(strPDF, strVSD)
    System.Diagnostics.Process.Start(strPDF)

Catch ex As Exception

    MsgBox(ex.Message)
```

End Try

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 Integer	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 = AppPath & "\debug.pdf"
Dim strDOC As String = AppPath & "\debug.doc"

Try

    oPDF.CreatePDFfromWord(strPDF, strDOC)
    oPDF.GarbageCollect()
    System.Diagnostics.Process.Start(strPDF)

Catch ex As Exception

    MsgBox(ex.Message)

End Try
```

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

Requirements: Access and file PDFSEND.EXE

Arguments:

strPSFile As String	Postscript file to create
strDatabase As String	Access database to use
strReportName As 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"

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

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"

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
objExcelWorksheet As Object	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 Integer	Page header alignment. Values: 0=left, 1=center, 2=right
Optional strFooter As String	Page footer
Optional lngFooterAlignment As Integer	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 Integer Beginning page number

Example:

```
Dim strPS As String = AppPath & "\sample.ps"
Dim strWB As String = AppPath & "\sample.xls"
Dim strRange As String
'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

Try

    oPDF.CreatePSfromExcel(strPS, strWB, 1, strRange, "", "", "", False,
False, "Page &P", PDFCreate.AlignType.pdoAlignRight, "&D &T",
PDFCreate.AlignType.pdoAlignCenter, 1.0, 0.5, 0.0, 1.0, 0.0, 0.0, 1)
    'oPDF.CreatePDFfromExcel(strPDF, strWB, "xtreme", strRange)
    oPDF.GarbageCollect()
    MsgBox("PS file created.", MsgBoxStyle.Information)

Catch ex As Exception

    MsgBox(ex.Message)

End Try
```

CreatePSfromImageFile() - Method to create a PostScript (PS) file from an image file. Note: If arguments intX or intY are omitted the image file will be resized to fit the page.

Requirements: File PDFSEND.EXE

Arguments:

strPSFile As String	Postscript file to create
strImagefile As String	Image file to use
Optional intX As Integer	X position
Optional intY As Integer	Y position
Optional intWidth As Integer	Width in pixels
Optional intHeight As Integer	Height in pixels
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"
Optional strSaveWindowCaption As String	Window caption. Default: "print to file"

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 intStartSlide As Integer = -1	PowerPoint start slide to use. Use -1 for first slide
Optional intEndSlide As Integer = -1	PowerPoint end slide to use. Use -1 for last slide
Optional blnHorizontalOrientation As Boolean	PowerPoint slide orientation. Values: True,False
Optional blnBlackAndWhite As Boolean	Black and white?

CreatePSfromPub() - Method to create a PostScript (PS) file from a Publisher file.

Requirements: Publisher

Arguments:

strPSFile As String	Postscript file to create
strPubFile As String	Publisher file to use
Optional strPrinter As String	Print driver. Default: "Generic PostScript Printer"

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
Optional blnFontBold As Boolean	Font bold. Values: True,False
Optional intFontSize As Integer = 8	Font size
Optional lngFontColor As Integer	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 Integer	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

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"

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"

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 Integer	Page number alignment. Values: 0=left, 1=center, 2=right
Optional blnFirstPage As Boolean	Add page number to first page? Values: True,False

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

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

Arguments:

strURL As String	URL of file to download
strFile As String	Filename to save to

Example:

'Note: you must modify the remote host settings before this will work
oPDF.DownloadFile <http://www.mysite.com/file.pdf>, "c:\temp\file.pdf"

GarbageCollect() – Method to release objects from memory.

Print() – Method to print any file (including PDF files) silently.

Arguments:

strFile As String	File to print
Optional blnWindow As Boolean	Create a new process window? Default is True
Optional intWaitForExit As Integer	Milliseconds to wait before closing the process window. Default is 10000 which is 10 seconds.

Example:

```
Dim strPDF As String = "C:\Temp\file.pdf"
oPDF.Print(strPDF)
```

PrintAPI() – Method to print any file (including PDF files) silently. Note: Uses Window’s API.

Arguments:

strFile As String	File to print
-------------------	---------------

Example:

```
Dim strPDF As String = "C:\Temp\file.pdf"
oPDF.PrintAPI(strPDF)
```

ResetPDFProperties() – Method to reset PDF property settings (PDFPassword, PDFPermissions, PDFVersion) to their default values.

Example:

```
oPDF.ResetPDFProperties()
```

SetBookmarks() – Method to add bookmark data to a PostScript file.

Arguments:

ByVal strPSFile As String	PostScript file to use
ByVal arrBookmarkArray As Array	Two dimensional array of bookmark data
ByVal blnShowBookmarks As Boolean	Show bookmarks when PDF is opened?
ByVal intInitialPage As Integer	Page to go to when PDF is opened
ByVal intInitialPageX As Integer	Page X coordinate to move to when PDF is opened

SetDocumentInfo() – Method to set document properties (Author, Subject, Title, Keywords) in a PostScript file.

Arguments:

ByVal strPSFile As String	PostScript file to use
ByVal strTitle As String	Document title
ByVal strAuthor As String	Document author
ByVal strSubject As String	Document subject
ByVal strKeywords As String	Document keywords

Example:

```
Dim strPS As String = "c:\Temp\file.ps"
Dim strPDF As String = "c:\Temp\file.pdf"

'Set some document properties
oPDF.SetDocumentInfo(strPS, "My PDF File", "SkySof Software Inc.",
"Subject goes here", "keyword1 keyword2 keyword3")

'create a PDF file from the PostScript file
oPDF.CreatePDFfromPS(strPDF, strPS)
```

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

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

Arguments:

strURL As String	URL to upload to
strFile As String	Filename to upload

Example:

```
oPDF.UploadFile "http://www.mysite.com", "c:\temp\file.pdf"
```

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

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

SkySof Software Inc.

<http://www.skysof.com>

Email: kusluski@bellsouth.net