

MAKEPDF.EXE - A front end utility for the AFPL Ghostscript program (previously Aladdin Ghostscript) to automate the creation of Portable Document Format (PDF) files. The program takes a postscript file and creates a PDF file from it. It defaults to creating the new file in the same directory with the same rootname and a .PDF extension.

and it's companion programs:

MAILPDF - A utility to automatically email a PDF document using any MAPI compliant mail system such as Pegasus Mail (or even MS Outlook) and

FTP-PDF - A utility to create a PDF document and then upload it to a Website

(MAILPDF and FTP-PDF are separate downloads available from the MAKEPDF page at <http://www.lexacorp.com.pg>)

Written in PB/DLL ver 6.1 (<http://www.powerbasic.com>)
by Stuart McLachlan
Lexacorp Ltd
PO Box 686, Boroko, Papua New Guinea
<http://www.lexacorp.com.pg>
Email: software@lexacorp.com.pg

MAKEPDF

=====

ver History

v1.0 Original Concept

v1.1 Improved Long File Name support

No longer restricted to using .PRN files. Accepts a filename with any or no extension

If invoked from the commandline with a -V1.2 or -V1.3 parameter, creates the PDF

without opening a dialog box and exits without any messages.

V1.1b Support for multiple periods in long file names. Only strips a trailing 3 character

or less extension.

V1.1c Display error message if path and filename is too long

V1.2 Option to automatically delete the input file on completion.

Option to specify the output file name.

Creates a new directory for the output file if necessary.

-Q (quiet) command line switch to suppress error messages.

V1.2a Minor Bug fix

V1.2b Added delete any previous PDF of same name and check to see whether new PDF file

actually created by Ghostscript.

V1.2c Minor Bug fix

V1.2d Minor Bug fix

V1.2e Added check to make sure that GSPROG does point to an executable

V1.3 Option to automatically view PDF after creation (uses default Windows PDF viewer)

-S command line switch ("Show") to select this option

V1.3a Bug fix in creating Ver1.3 PDF with view option selected

V1.3b Added check for input and output files being the same name.

V1.4 Added browse button to select input file.

V1.5 Accepts wildcards for the input specification in Interactive mode

-L switch to force the output file name to be all lower case

AutoExit switch for interactive mode

V1.5a Bug fix - autoExit only worked if autoShow selected.

V1.5b Bug fix for commandline passing differences between W9X and Nt/W2K

V1.6 Added a Directory Browse button for selecting an output directory

Added Option to make the Output Directory field "sticky" when selecting input files

V1.7 Replaced the Ghostscript Environment variable with an Ini file to simplify set up.

Program prompts for the location of Ghostscript when run for the first time.

V1.8 Creates a temporary file for the parameters to pass to Ghostscript, allowing an

increase in the length of filenames to 123 characters each.

Modifications to create v1.4 PDF (requires at least AFPL Ghostscript ver 6.50)

V1.8a Bug fix - to correct problem passing temporary file with spaces in pathname

V1.8b Bug Fix - to correct occasional upper/lower case problem with extension of output file

V1.9 Added -M switch to send resulting PDF file to MAPI compliant mail client from the command line

V1.9a Bug fix - to correct "invalid switch" message with hyphens in filenames passed from the command line

V2.0 If MakePDF is installed in a R/O directory (for instance on a network), the

MakePDF.INI

file will be placed in the users Windows directory.

The temporary parameters file is now created in the users Temp or Tmp directory

rather

than the MakePDF directory.

The -M Switch can optionally pass a Subject line for the email.

V2.1 Changed switch delimiter from " -" to " /"

V2.2 Separated Output File and Directory boxes

Changed to single "MakePDF" button with PDF Versions as options buttons and a default to v1.4

Prompts to overwrite if PDF exists (unless No Message flag is set)

Using MakePDF

=====

1. Download and install the Ghostscript application. (<http://www.cs.wisc.edu/~ghost/>)

2. Set up a new printer in Windows using any available Postscript printer driver and set the port to "FILE:" (preferably a colour printer - I used the HP Paintjet XL300 Postscript and renamed it "PostscriptFile"). You can also set the "Spool Settings" to "Print Direct To Printer". See below - "Variations on a theme" for more on directing the printer to a file

3. Place MAKEPDF.EXE in any convenient directory.

You can use MakePDF on a postscript file in a number of ways:

a. Put shortcut to MakePDF in "C:\WINDOWS\SENDTO" and then right click on a postscript file and use the Send To Menu.

b. Place a shortcut to MakePDF on the desktop and drag and drop files onto it.

c. Specify MakePDF as the default "Open" program for a particular file extension (in Windows Explorer: go to View - Options - File Types - New Type), name your postscript files appropriately and then double click on any postscript file.

d. Call MakePDF from the commandline or from a batch file with the postscript filename as an argument.

To Create PDF documents:

=====

1. Create your document in the application of your choice (MS Word, Excel or whatever)

2. Print the document to the PostscriptFile printer (if you have set up MakePDF as a default "Open" application, use the appropriate filename extension).

(Note the total length of the path+filename you can use is limited to 123 characters.)

3. Locate the resulting Postscript file and double click, right click and select MakePDF or drag onto the PDF icon. Or open MakePDF and use the "Browse" button to locate your Postscript file.

4. A dialog box appears, click on a button to create a Ver 1.2 or VER 1.3 PDF file (Ver 1.2 for Acrobat 3 or later, 1.3 for Acrobat 4 or later)

You can also invoke MakePDF from the commandline or a batch file. In this case the usage is "MAKEPDF filename [-OutputFileName] [-V1.2|-V1.3]". If you include -V1.2 or -V1.3 after the filename, MakePDF will automatically create the appropriate PDF file and then terminate without any dialog boxes or user interaction.

This program is copyright Lexacorp Ltd, April 2000 - June 2001
It may be freely distributed.

This program is boozeware. If you find it useful and enjoy a beer, buy me one.
Since I won't be there, drink it for me
(if you don't drink alcohol - have an icecream or something)!

VARIATIONS ON A THEME

=====

Instead of sending your Postscript printer to File: you can send output to a specified file each time. YOU can do this in two ways:

1. Edit WIN.INI and create a new entry under [Ports] consisting of a filename followed by a "=" (without the inverted commas).
2. Select "Add Port" in the printer properties of any configured printer, click on Local Port, click OK and enter the filename.

If you then select this file as the printer port, it will be created each time you print to it (any old file will be deleted). This will avoid having to enter a filename everytime you create the Postscript file. In this case you may want to use the -O option to change the name of the output file.

In my case, I have directed the postscript printer to C:\WINDOWS\DESKTOP\PRINTFILE.PSF

I have then created a "Postscript File" association for the .PSF extension

and created an OPEN action with a command of:

```
"C:\MY DOCUMENTS\SOURCE\MAKEPDF\MAKEPDF.EXE" %1 -D -V1.2 -OC:\WINDOWS\DESKTOP\PDFfile.PDF
```

This means that when I print a document, it appears on the desktop.

When I double click on

it a PDF file appears and the temporary file is deleted.

If you include the -M switch, once the file has been created, it will open your MAPI

compliant mail program to a new message with the PDF file already attached - optionally with

a specified email Subject. (If you want to include addresses, message body etc so that the

message is actually sent automatically - use MailPDF instead - see below)

You can create a macro in MS Word, Excel etc to automate this process.

For MS Word, you can place the following macro in your Normal.DOT template and place a

button pointing to it on a suitable toolbar. Then you can print the current document as a

PDF and open a new mail message with it already attached - all with a single mouse click.

```
Sub SendPDF()
```

```
    Dim strShellstr As String
```

```
    Dim strQ As String
```

```
    'set strQ as a Quote character to delimit long file names containing spaces
```

```
    strQ = Chr$(34)
```

```
    'Print to Postscriptfile
```

```
    ActivePrinter = "Postscriptfile on C:\WINDOWS\TEMP" 'set the appropriate printer
```

```
    ActiveDocument.PrintOut (False) 'False switch forces macro to wait until file is printed
```

```
    'Create PDF using MakePDF
```

```
    'passes the documents "Title" property as the Subject of the email
```

```
    'and the document's Name (with a .PDF extension instead of the .DOC extension)
```

```
    'as the filename
```

```
    strShellstr = strQ & "C:\Program Files\makepdf\makepdf.exe" & strQ &
```

```
        "C:\WINDOWS\DESKTOP\PSSCRIPT.PSF -D -V 1.2 -M " & _
```

```
        ActiveDocument.BuiltInDocumentProperties(wdPropertyTitle) & " -Q -O " & _
```

```
        strQ & "C:\windows\desktop\" & _
```

```

    Left$(ActiveDocument.Name, Len(ActiveDocument.Name) - 4) & ".PDF" & strQ
    Shell strShellstr

End Sub

```

If you wish to update PDF files on a website, you can also automate this by using another of my utilities - FTPSend. FTPSend reads the Servername, Username, Password, RemoteDirectory and LocalFilename from the command line or a parameter file and uploads the file by FTP.

```

MAILPDF
=====

```

```

ver History
v1.0 Original Concept
v1.1 Modified with MakePDF v2 to allow alternat INI file and temp file directories

```

Once you have set up MakePDF as above, you can run MailPDF from the same directory as MakePDF to convert a document and email it. I wrote this one to automate sending reports generated in MSAccess, but it can also be use for any other application which can shell to a command line or from a batch file.

(Note there is little error checking built in - it's up to you to make sure that the parameters you pass are correct!)

MailPDF takes the following parameters:

```

-A the email address (or addresses separated by semicolons) to send the report to if using MS Outlook
-N The email address(es) for any other MAPI complaint mail program or the Address Book name(s) if using MS Outlook
-I the input name of your postscript file (in my case, I send postscript output automatically to a file on the desktop called PSCRIPT.PSF
-O the name of the final PDF file to send (in the example, I create a PDF file called C:\MyReport)
-S The Subject line of the email message
-B the body of the email

```

As a simple example in MS Access:

1. Design a report ("MyReport") and configure it to "print to a spec

ific printer" -
specifically, the printer set up for MakePDF to print to a specific file.

Create a Form with a text field to enter the email address (txtAddress) and a command button (cmdSend) and place the following code in the on_click event:

```
cmdSend_click()
DIM strShellstring as string
DoCmd.OpenReport "MyReport"
strShellstring = Chr$(34) & "C:\Program Files\Makepdf\Mailpdf.exe" &
  Chr$(34) & " -N "
  & txtAddress & " -I C:\Windows\Desktop\Pscript.psf" & Chr$(34)
  & " -O C:\MyReport -S Attached Report -B Here is the report you
  wanted"
Shell (strShellstring)
End Sub
```

Clicking the button will print the report, convert it to PDF and send it to the specified address.

Note:

Because of the spaces in the path, it is necessary to surround the MailPDF.exe path in the string with quotes (Chr\$(34)'s)

FTP-PDF =====

Once you have set up MakePDF as above, you can run FTP-PDF from the same directory as MakePDF to convert a document and upload it to a web site. Use it in a similar way as MailPDF in macros or procedures of your document creating application

ver History
ver 1.0 Original Concept

FTP-PDF takes the following parameters:

```
/C parameterfilename (Create parameterfile - use full path)
/R parameterfilename (Read parameterfile - use full path)
/S Server
/U Username
/P Password
/D remoteDirectory
/F postscriptFilename (the file your printer device sends output to
- use full path)
/O Outputfilename (the name you want to give to the file on your w
```

ebsite, must end in
.pdf"

If the /C switch is used, the remaining parameters will be saved encrypted in

ParameterFilename (no FTP transfer will take place!).

If the /R switch is used, the parameters will be READ from the encrypted ParameterFilename

which must be created first by running FTP-PDF with the /C switch.

If any other commandline parameters FOLLOW the /R switch they will over-ride parameters in

the parameter file.