Yash Honda PDF (`yhpdf`) Manual

Nebhrajani A.V.

July 20, 2021

Contents

1: Installation

1.1 Basic

Use MS Windows 10.

1. Install **Ghostscript (64-bit)**:
   (a) Go to `https://www.ghostscript.com/download/gsdnld.html`.
   (b) Download the file **Ghostscript 9.54.0 for Windows (64 bit)** with the AGPL license (first column). Do not use the 32 bit version, it is incompatible with `yhpdf`.
   (c) Run the installer with the standard steps, and click `Finish` once done. It should launch a website, telling you Ghostscript is installed.
2. Get **yhpdf**:
   (a) Download `yhpdf.exe` from `https://nebhrajani-a.github.io/yhpdf/yhpdf.exe`.
   (b) Copy `yhpdf` from your `Downloads` folder and paste it in `C:\Program Files` (This PC → Local Disk → Program Files).
   (c) Double click on `yhpdf`. Wait for 10 seconds. If it doesn't launch, double click again, and wait for 10 more seconds. On old machines, `yhpdf` takes time to start up. Be patient, it runs much faster than it starts.
   (d) Once it starts, you should see a windows with four buttons and some text. Go down to the taskbar, right click on `yhpdf`'s feather icon, and select "Pin to taskbar".
   (e) If `yhpdf` doesn't start and reports an error, make sure Ghostscript is installed. If there's some other error you can't resolve, report it to the package maintainer.

1.2 Advanced

WIP.

2: Usage

Usage is intuitive:

1. Click “Select a file” to upload the PDF you want to split and compress.
   (a) A popup will tell you which file has been selected. Click “OK”.
2. Click “Select output folder” to select the folder into which the new PDFs will be put.
   (a) A popup will tell you which folder has been selected. Click “OK”.
3. Click “Process PDF” to process the PDF. This can take up to 60 seconds, depending on the speed of your machine and the size of the input PDF.
   (a) Once the PDF process is complete, a popup will ask you if you want to view the output files. Click “Yes” or “No”.
4. Process as many files as you want to by selecting them and the output folder.
5. Click “Exit” when you’re done.

3: Technical Description

3.1 What?

`yhpdf` is a GUI tool which:

- Takes an input PDF file.
- Splits it into one PDF file per page.
- Compresses each page PDF to less than 195KB.

Given the intended use-case, the only platform with first-class support is MS Windows 10 (compiled binary provided). Other operating systems may be supported on a per-case basis.

3.2 How?

`yhpdf` uses PyPDF2 to split the PDF page-wise, then uses Ghostscript's compression on each of these pages. If the file size is greater than 195KB, it first tries `-dPDFSETTINGS=/ebook`. If the file size is still greater than 195KB, it tries `-dPDFSETTINGS=/screen`. This gets most scanned pages below 200KB. If this fails, it raises an error. In essence, `yhpdf` is just a `for` loop which generates per-page PDFs then compresses them.

The GUI uses Tkinter and is cross-platform. If you're proficient in Python, you can probably get `yhpdf.py` running in your own environment by installing all dependencies.

3.3 Known Issues

`yhpdf` is a quickly written program, and will **not** win any software engineering competitions. It's meant to do one very specific job well. Thus, it has many unhandled exceptions, an ugly interface, and rather ugly code: but it works, and it works fairly fast.

The biggest user-facing issue is the slow startup time. This is unfortunately out of `yhpdf`'s control, since `pyinstaller` and all other Python packagers need to decompress the single `.exe` file to run, which takes time, especially on Windows. Screw you, Microsoft. Moreover, it includes within it an entire Python interpreter so…you know how that goes.

If this bothers you, either keep it running all the time by auto-starting it (it uses next to no RAM and CPU while idling), or get Python and install dependencies (needless to say: advanced users *only*).

4: yhpdf.py Source

```
import tkinter as tk
from tkinter import ttk
from tkinter import filedialog as fd
from tkinter.messagebox import showinfo
from tkinter.messagebox import showerror
from tkinter.messagebox import askquestion
import sys
import os
from pathlib import Path
import ghostscript
from PyPDF2 import PdfFileWriter, PdfFileReader

root = tk.Tk()
root.title("[Yash Honda] PDF Splitter/Compressor")
root.geometry('480x480')

list_of_vals = {'filename': False, 'dir': False}

def split_and_compress():
    try:
        inputpdf = PdfFileReader(open(str(list_of_vals['filename']), "rb"))
    except FileNotFoundError:
        showerror(title='Error', message="File not found!")

    size_limit = 197500
    for i in range(inputpdf.numPages):
        output = PdfFileWriter()
        output.addPage(inputpdf.getPage(i))
        with open("%s/doc-page%s.pdf" % (list_of_vals['dir'] ,i), "wb") as
        ↪  outputStream:
            output.write(outputStream)
        if (size_limit - int(os.stat("%s/doc-page%s.pdf" % (list_of_vals['dir'],
        ↪  i)).st_size)) < 0:
            args = """-dCompatibilityLevel=1.4 -dNOPAUSE -dBATCH -dQUIET
            ↪  -sDEVICE=pdfwrite -dPDFSETTINGS=/ebook
            ↪  -sOutputFile=%s/document-page%s.pdf %s/doc-page%s.pdf""" %
            ↪  (list_of_vals['dir'], i, list_of_vals['dir'], i)
            args = args.split()
            ghostscript.Ghostscript(*args)
            os.replace("%s/document-page%s.pdf" % (list_of_vals['dir'], i),
            ↪  "%s/doc-page%s.pdf" % (list_of_vals['dir'], i))
            if (size_limit - int(os.stat("%s/doc-page%s.pdf" %
            ↪  (list_of_vals['dir'], i)).st_size)) < 0:
                args = """-dCompatibilityLevel=1.4 -dNOPAUSE -dBATCH -dQUIET
                ↪  -sDEVICE=pdfwrite -dPDFSETTINGS=/screen
                ↪  -sOutputFile=%s/document-page%s.pdf %s/doc-page%s.pdf""" %
                ↪  (list_of_vals['dir'], i, list_of_vals['dir'], i)
                args = args.split()
                ghostscript.Ghostscript(*args)
                os.replace("%s/document-page%s.pdf" % (list_of_vals['dir'], i),
                ↪  "%s/doc-page%s.pdf" % (list_of_vals['dir'], i))
```

```
            if (size_limit - int(os.stat("%s/doc-page%s.pdf" %
            ↪  (list_of_vals['dir'], i)).st_size)) < 0:
                showerror(title='Error', message="ERROR: I couldn't compress page
                ↪  %s to less than 195KB. Consider using an online compressor or
                ↪  re-scanning." % (i+1))
    op = askquestion(title='Success', message='Successfully processed %s. Open
    ↪  output folder?' % (list_of_vals['filename']))
    if op == 'yes':
        os.startfile(list_of_vals['dir'])

def select_file():
    filetypes = (
        ('PDF files', '*.pdf'),
        ('All files', '*.*')
    )

    list_of_vals['filename'] = (fd.askopenfilename(
        title='Select scanned PDF',
        initialdir = str(Path.home()),
        filetypes=filetypes))

    if list_of_vals['filename']:
        showinfo(
            title='Selected File',
            message="Selected file %s" % (list_of_vals['filename'])
        )

def select_directory():
    list_of_vals['dir'] = fd.askdirectory(title='Select output folder', initialdir
    ↪  = str(Path.home()))

    if list_of_vals['dir']:
        showinfo(
            title='Selected folder',
            message="Selected folder %s" % (list_of_vals['dir'])
        )


text = ttk.Label(root, text="This is an internal program. You may not copy,
↪  distribute, or sell it.")
text.pack(expand=True)

open_button = ttk.Button(root, text='Select a file', command=select_file)
open_button.pack(expand=True)


dir_button = ttk.Button(root, text="Select output folder",
↪  command=select_directory)
dir_button.pack(expand=True)

process_button = ttk.Button(root, text="Process PDF", command=split_and_compress)
process_button.pack(expand=True)

exit_button = ttk.Button(root, text="Exit", command=sys.exit)
```

```
exit_button.pack(expand=True)


text2 = ttk.Label(root, text="All rights reserved.\n(C) Yash Honda 2021")
text2.pack(expand=True)

root.mainloop()
```