PDF Shrinker Documentation

Chinmaya S Rathore

Version 1.0.0

October 2025

PDF Shrinker Documentation

Table of Contents

1. Introduction

PDF Shrinker is a minimalistic and easy to use GUI app for compressing PDFs. It can also convert colored PDFs to grayscale and flatten them. It works on top of Ghostscript which needs to be installed on the users computer. Ghostscript is *open source and free* free for non-commercial use.

Using PDFShrinker, users can significantly compress or shrink PDFs as per their requirements. Typical use cases include sending large PDF documents by email, uploading PDF documents on sites that have PDF upload file size restrictions like visa applications, college applications or government document submission or just generally optimizing your PDFs for acceptable file sizes as per intended use.

Compression works best when PDFs have images

It is worthwhile to note that a PDF that has many images benefits the most from file compression. In contrast, if the PDF contains all or mostly text, compression gains may not be substantial.

2. Prerequisites

2.1. Install Ghostscript

2.1.1. Window Users

For windows, download and install Ghostscript from the Ghostscript download page (Figure 1) and then double-click the downloaded executable file (gsxxxw64.exe for 64-bit or gsxxxw32.exe for 32-bit) and follow the installation wizard's prompts.

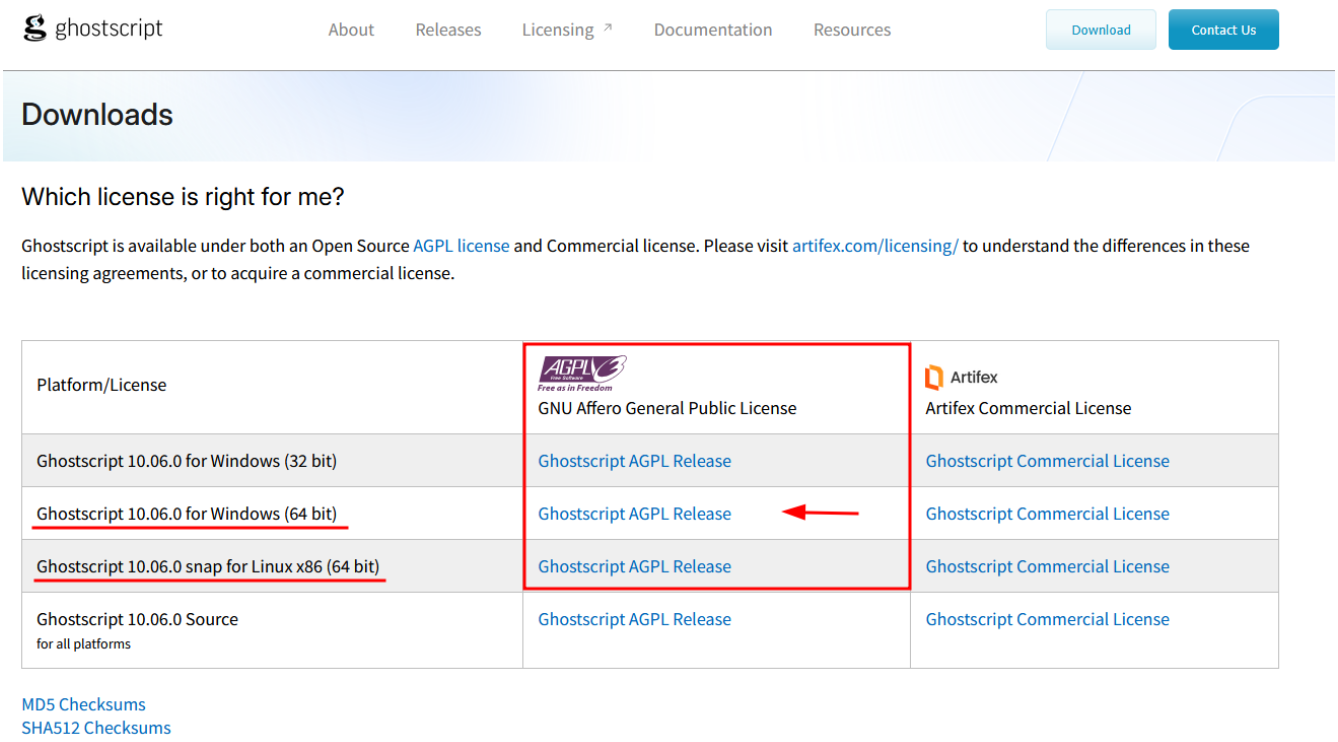

Figure 1. Ghostscript Download Page with download options for Windows 32 bit, 64 bit and Linux (Snap)

2.1.2. Linux Users

Linux users, can first check if they have Ghostscript already installed by using the following command from the terminal:

```
ghostscript --version

or

gs --version
```

If no version number is reported, use the following command from the terminal to install Ghostscript:

Ubuntu/Debian

Install Ghostscript via the snap package on the Ghostscript download page or from the terminal command line using the following commands:

```
sudo apt install ghostscript
```

To test successful installation, type the following command from the terminal:

```
gs --version
```

RHEL/Fedora

Install Ghostscript from the terminal using the following commands:

```
sudo dnf install ghostscript
```

3. Download and Install PDF Shrinker

Once you have Ghostscript installed on your computer, you can download and install PDF Shrinker from the PDF Shrinker Download Page for your OS (Windows or Linux). Please see Appendix A if you wish to compare SHA256 checksums for the downloaded package.

3.1. Windows Users

Download PDF Shrinker installer *pdfshrinker_1.0.0_x64-setup*.exe, double click on the downloaded file and install following screen instructions.

Windows users, while running the installer, may get a system advisory during installation (Figure 2). They will have to choose `Yes` to proceed with the installation. This message is generated because the app has not been digitally signed by the author. App signing certificates are quite expensive !

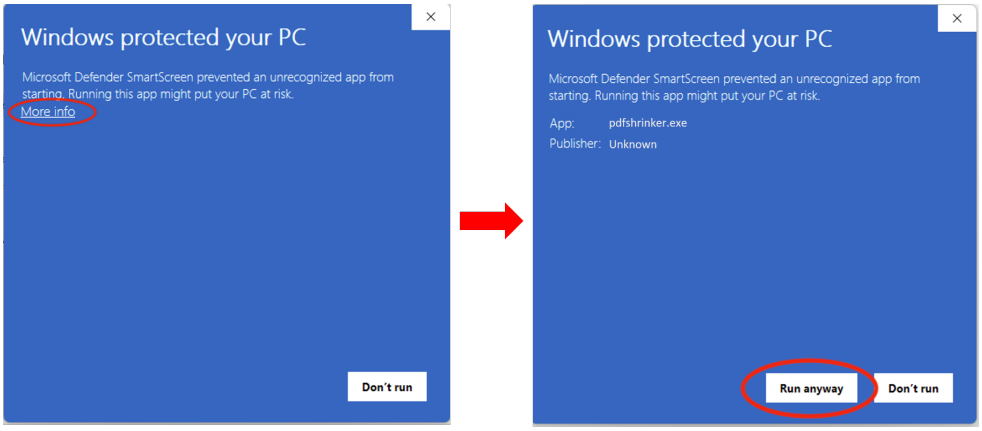

Figure 2. Unknown Publisher Message for Windows Users

As PDF Shrinker is open source, anyone can inspect the source code from it's project page.

If you start PDF Shrinker without installing Ghostscript, a message (Figure 3) requesting you to install ghostscript will be displayed and the app will exit when the `OK` button is pressed.

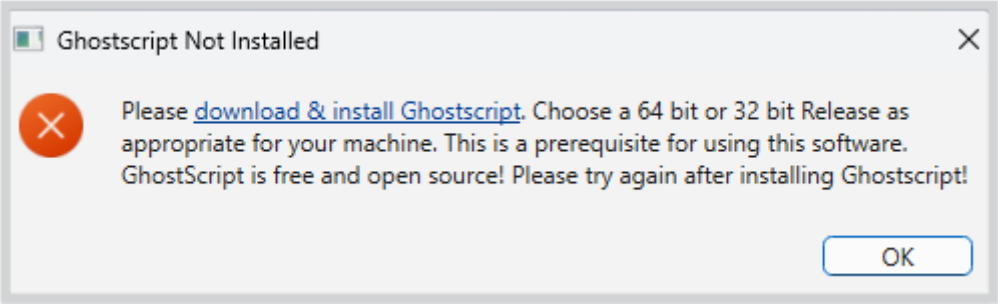

Figure 3. Install Ghostscript before running PDF Shrinker

The user should go to the Ghostscript download page and download and install Ghostscript and then retry running PDF Shrinker. This is a one time install.

About Ghostscript

Ghostscript is a command line interpreter for the PostScript® language and PDF files. It is developed and maintained by by Artifex Software, Inc. and has been under active development for over 30 years. It is open source and free for non-commercial use and is widely used. Ghostscript has an extensive command set which is detailed in the Ghostscript documentation.

3.2. Linux Users

3.2.1. Ubuntu/Debian

Download the *pdfshrinker_1.0.0_amd64.deb* package and double click on the downloaded file to install via the software manager.

OR

open a terminal, navigate to the folder that has the downloaded *pdfshrinker_1.0.0_amd64.deb* package and issue the following command:

```
sudo dpkg -i pdfshrinker_1.0.0_amd64.deb
```

3.2.2. RHEL/Fedora

Download the *pdfshrinker_1.0.0_amd64.rpm* package and double click on the downloaded file to install via the software manager.

OR

```
open a terminal, navigate to the folder that has the downloaded
```

_pdfshrinker_1.0.0_amd64.rpm_ package and issue the following command:

```
sudo dnf install pdfshrinker_1.0.0_amd64.rpm
```

4. Using PDF Shrinker

PDF Shrinker is quite easy to use. It has been designed with simplicity and user convenience in mind. Essentially, the user just has to provide the input PDF that they wish to compress, retain the defaults or choose compression and compatibility settings (see Table 1 and Table 2) and press `Shrink PDF` button to compress the file. The user can also further tweak the level of compression using *advanced options* (described later in Section 5). That's it.

When you start PDF Shrinker, you will see the GUI shown in Figure 4.

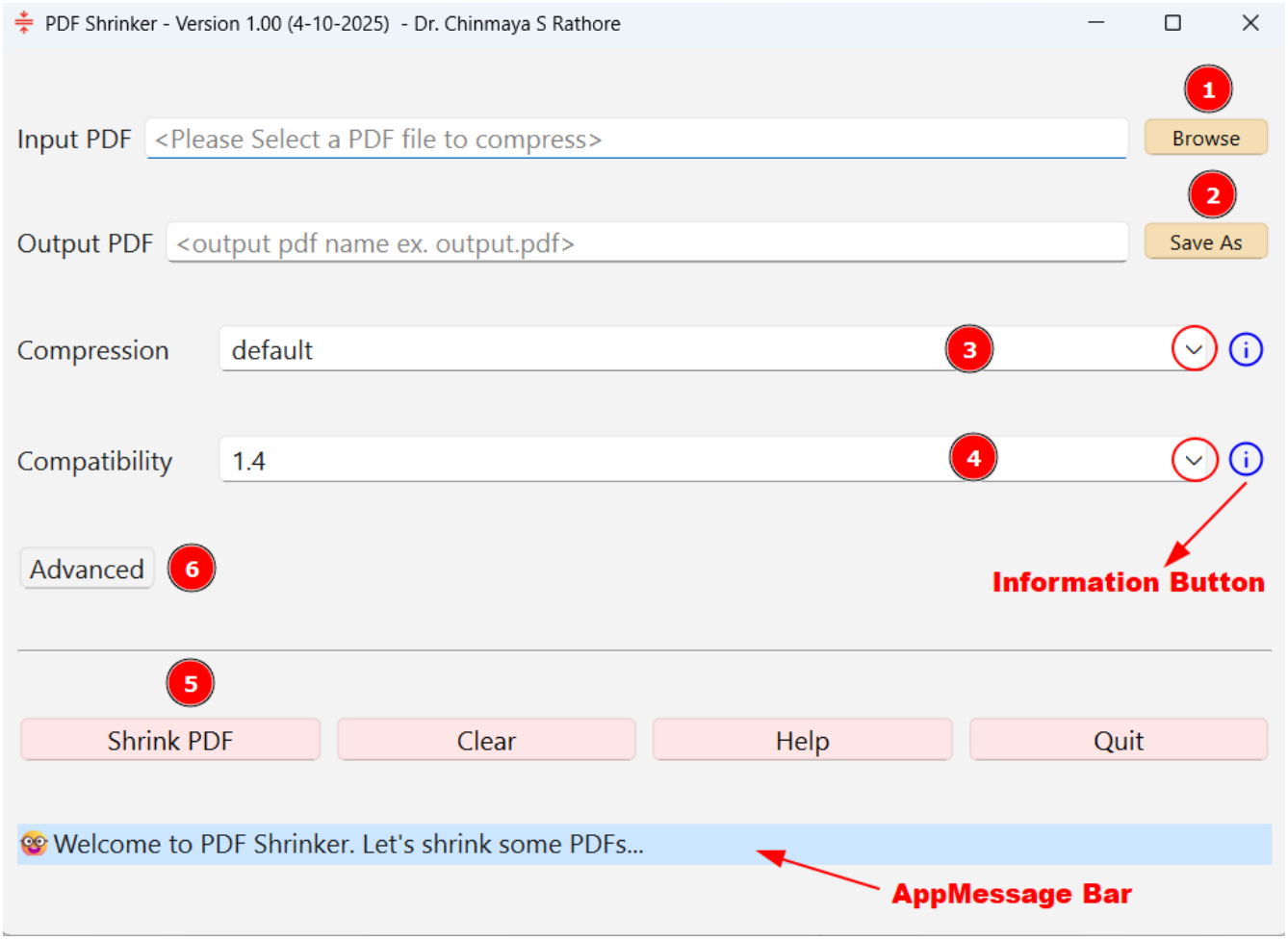

Figure 4. PDF Shrinker GUI

Referring to the PDF Shrinker GUI (Figure 4), the user can follow the sequence below to compress a PDF file:

1. Choose a PDF file to compress by using the `Browse` button for the *Input PDF* (1).
2. The user can the provide a name for the output file i.e. the compressed PDF file using the `Browse` button for the *Output PDF* (2).

3. The user can now choose a compression level using the dropdown in (3). Different compression presets that can be chosen are explained in Table 1.

Table 1. Compression presets and PDF quality

Preset	Purpose	Image Resolution	Compression and Quality
default	A versatile, all-purpose setting that provides a balance between file size and output quality.	Varies, typically a medium-to-high resolution.	Good overall quality with moderate compression.
screen	Creates the smallest file size possible, optimized for on-screen viewing.	Lowest resolution (72 dpi).	Highest compression, which results in the lowest quality output.
ebook	Creates a medium-sized file with a medium-quality output, suitable for distribution as an e-book or for general use.	Medium resolution (150 dpi).	Balanced compression and quality.
printer	Creates a high-quality PDF, suitable for printing on desktop printers or digital copiers.	High resolution (300 dpi).	Lower compression to maintain high image quality.
prepress	Creates the highest-quality file for professional printing and prepress workflows.	High resolution (300 dpi) with color preservation.	Largest file size with the least compression to preserve all image and color data.

4. The user can now choose a compatibility setting (4) (or use the default 1.4). What these presets mean is summarized in Table 2. Usually 1.4 and 1.5 lead to good compression but you can try others as well to see if they work for you. While the default is 1.4 which is considered a safe setting for new and old viewers and printers, choosing a higher compatibility setting works with all web browsers and most popular PDF viewers.

Table 2. Compatibility Presets and their Meaning

Value	PDF Specification	Year	Description
1.3	Acrobat 4	2000	Maximum legacy support, best for older viewers and broad access. Removes transparencies and flattens the PDF.
1.4	Acrobat 5	2001	Safe default; supports transparency and widely compatible.
1.5	Acrobat 6	2003	Adds object streams and more compression features.
1.6	Acrobat 7	2005	Better support for transparency, encryption, and advanced features.

Value	PDF Specification	Year	Description
1.7	Acrobat 8	2006	Broader feature set, improved security and compression.
2.0	Acrobat 10	2020	Modern features, compression, only for new viewers.

5. Press the `Shrink PDF` button (5) to start PDF compression. You will get a *working* message indicating that the software is compressing the inputPDF in the app message bar as shown in Figure 5(1).

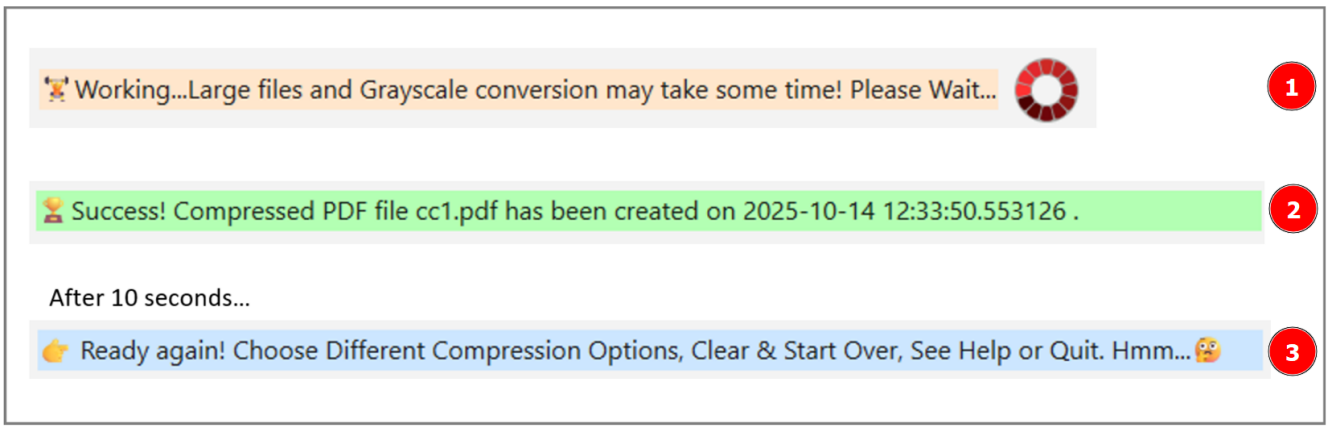

Figure 5. Compression activity and success messages on the app message bar

6. When the task is complete, you will get a message showing that the job has been successfully completed (Figure 5(2)) which will after 10 seconds change to the 'Ready again' message shown in Figure 5(3). If you are trying to compress a large file, it may take a while to complete the task. So please wait for the activity spinner circle to stop. During the period that the compression is underway, the `Shrink PDF` , `Clear` and the `Quit` buttons are deactivated to ensure that the user does not start another process or quits the app while the process is running.
7. You can now open the output file folder and examine the compressed PDF file. If the figures, tables and graphs in the PDF are not of acceptable quality, try a different preset (like ebook, or default) or click on the `Advanced` button to control image resolution while compressing.

Other buttons seen next to the `Shrink PDF` button work as follows:

- The `Clear` button clears all the fields and resets the screen to the initial state.
- The `Help` button opens the project page on GitHub from where the user can read or download documentation (this document!) and watch a video tutorial on how to use PDF Shrinker.
- The `Quit` button exits or closes the app.

5. Advanced Settings

The advanced settings provide fine grained control on quality and size. Clicking on the `Advanced` button reveals the advanced options as shown in Figure 6.

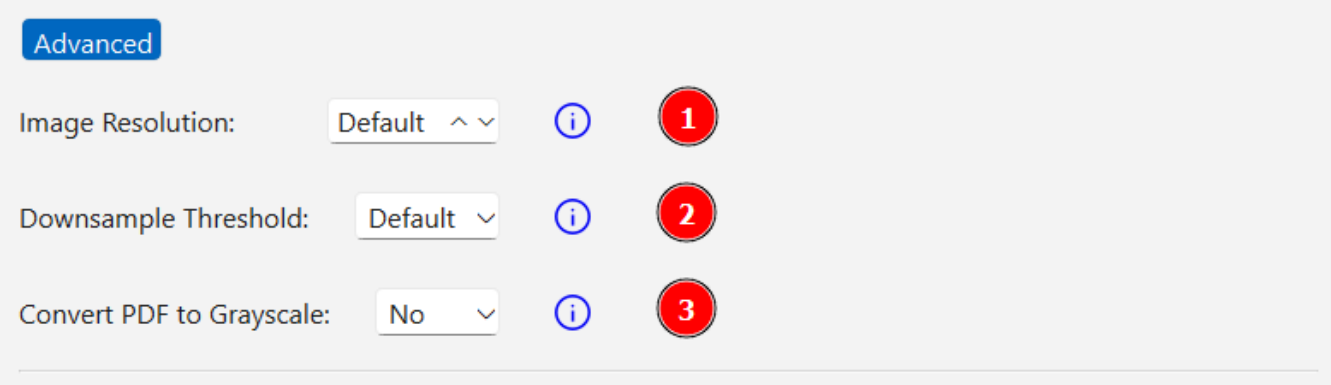

Figure 6. Advanced Options

These options are explained in the subsequent sections.

5.1. Image Resolution Setting

You can choose an image resolution setting between 50-200 by manipulating the image resolution spinner (1). This setting helps you override the resolution setting on the compression preset and increase or decrease the resolution of images in the compressed output PDF. A typical case on how this can be used with compression presets is illustrated in Example 1.

Example 1. A Typical Use Case

Let's say a PDF that we are trying to compress has an original size of 10 MB. If we compress using the *ebook preset* and 1.5 compatibility settings (without using advanced options), we get a compressed file that is 8 MB in size. From Table 1 it can be seen that ebook resamples images in the original PDF at 150 DPI resolution. What this means essentially is that an image that was at say 300 DPI resolution in the original PDF will now be down-sampled at 150 DPI in the compressed image. If however you find that the compressed size of 8 MB is not what you need and wish to reduce it further, you can set the image resolution in the advanced section to say 125 or 100 and press the `Shrink PDF` button. This will result in a file say at 5 MB keeping all other options of the ebook preset intact. Conversely, if you want to improve the quality of the images in the compressed PDF say using a ebook preset, you can also choose to set the resolution to 175 DPI. The same can also be done to override the *screen* preset (or other presets for that matter) which downsamples images to 72 DPI (Table 1). You can set the compression preset to *screen* and then in the advanced settings, choose an image resolution of say 90 or 100 DPI to improve the quality of images in the compressed PDF.

5.2. Downsample Threshold Resolution Setting

1. The 'Downsample Threshold' setting [Figure 6 (2)] can be used quite effectively in conjugation to the image resolution setting in the advanced section.

5.2.1. How Downsample Threshold Works

The purpose of downsample threshold setting is to control the aggressiveness of the downsampling process for color images. The most useful values are between 1.0 and 2.0 with the default being 1.5.

Interpretation of values

The threshold value is multiplied by the target resolution to determine the cutoff point. Only images whose effective resolution is higher than this calculated threshold will be downsampled. For example,if the image resolution=150 and the downsample threshold value is the default 1.5, the threshold dictates which images are downsampled as per the calculation shown in Example 2:

Example 2. Interplay of down sample threshold and image resolution

The downsample threshold value of 1.5 is multiplied with the image resolution value of 150:

```
1.5 * 150 = 225 dpi
```

This effectively means that all images in the original PDF with a resolution higher than 225 dpi will be downsampled to 150 dpi. Images below 225 dpi will be left untouched.

Similarly for a downsample threshold value of 1.0 the calculation is : 1.0 * 150 dpi = 150 dpi

What this means is that all images with an effective resolution higher than 150 dpi will be downsampled. As can be seen a downsample threshold value of 1 is the most aggressive setting for downsampling and results in the smallest file size.

Choosing a downsample threshold value of 2.0 in the above example works out to be as follows:

```
2.0 * 150 dpi = 300 dpi
```

Only images with a very high resolution (over 300 dpi) will be downsampled. This is a conservative setting that preserves the quality of most images.

Table 3 provides some broad practical guidelines for interpreting and choosing downsample threshold values.

Table 3. Practical guide to choosing downsample threshold values

Downsample Threshold	Compression	Effect on Images
1.5 (Default)	Balance between file size and quality	Downsamples images that are excessively high-resolution.
1.0	Maximize file size reduction	ensures that almost all color images are reduced to the specified target resolution.
2.0	Minimize quality loss-Less compression	Only extremely high resolution images are downsampled.

Downsample Threshold	Compression	Effect on Images
Greater than 2	Prevent downsampling	Useful if you want to apply other compression techniques without affecting image resolution.

5.3. Convert PDF to Gray scale

Use this setting to convert a coloured PDF to black & white (actually gray scale) PDF.You can use it with image resolution and downsample threshold to downsample any gray scale images in the original PDF that you are compressing. Converting a PDF to gray scale significantly reduces the PDF size.

Table 4 Provides a summary of some possible use cases for gray scale PDFs.

Table 4. Some Use Cases for Gray Scale PDFs

Scenario	Why Grayscale Is Preferred
Legal filings	Reduces file size and meets court-specified formatting rules.
Academic papers	Most journals require black-and-white figures to keep publications uniform.
Internal reports	Saves storage on corporate servers and speeds internal distribution.
High-volume printing	Cuts ink costs and eliminates color-related printing errors.
OCR-heavy workflows	Improves text-recognition accuracy and processing speed.
Accessibility	High-contrast grayscale can be easier for screen-reader software to interpret when paired with proper tagging and alt-text.

6. Flattening PDFs

If you are trying to upload a book PDF on Kindle Direct Publishing, you might be asked to *flatten* your PDF or remove all transparencies. Transparency refers to the degree to which an object or material allows light to pass through it. In digital graphics, it is controlled by a value called the alpha channel. A simple and practical example of transparency is a logo saved in the PNG (Portable Network Graphics) format. If you have a company logo with a white background and you save it as a JPEG, the white background is a permanent part of the image. When you place it on a webpage with a different color, like blue, the logo will appear in a white box. If you save the same logo as a PNG with a transparent background, the background is removed, and the image only consists of the logo itself. When you place this transparent PNG on the same blue webpage, the logo's edges will blend seamlessly with the background.

With PDF shrinker, you can flatten PDFs. To *flatten*, use the following steps:

1. Choose a compatibility setting of **1.3**. A compatibility setting of 1.3 removes all transparent elements in the PDF.
2. You can either leave the compression setting to **default** (for a smaller file size) or choose the **prepress** to preserve the original resolution of images (larger file size). You can also try , **printer** compression setting or other compression settings if you wish.
3. Press the `Shrink PDF` button

If the **printer** or **prepress** settings are chosen for compression, the resulting file will retain a very high quality of images but its size might be *larger* (even double in some cases) as compared to the original. This is because images with transparency and vector graphics in the original PDF are rasterized (bitmap or converted into a grid of pixels). Form fields if any in the PDF also become non editable. In summary, layers in the PDF are merged into a single layer or *flattened*.

7. Some Broad Guidelines for Compressing PDF file

Compressing or reducing the size of PDF files should be approached with an experimental mindset where the user can try various presets with or without advanced settings to get the desired compression that is acceptable in terms of size and quality. Some broad guidelines that might be helpful are as follows:

1. PDF files that have coloured images will be most benefitted by compression. If you have a text heavy PDF file with very few images, the compression might not be substantial.
2. First start with the *default* compression and compatibility settings and see if the resulting PDF file meets your requirements. This operation just involves you to provide the input and output PDF file names and press the `Shrink PDF` button. It is the fastest way to compress and the output should both read well on screen and produce acceptable printed copies even on older printers.
3. If the above does not meet your compression needs, try the *ebook* preset with 1.4 or 1.5 compatibility setting and press the `Shrink PDF` button. This should typically produce acceptable PDFs for screen reading and printing.
4. If you still need a smaller sized file, keeping the ebook preset and 1.5 compatibility, choose a image resolution of say 125 (you can try 100 or other intermediate resolutions) and a downsample threshold of 1 from the advanced setting and press the `Shrink PDF` button. Recall that ebook downsamples images to 150 DPI and to make the file smaller, we are reducing the resolution from 150 to 125 via this operation. This should result in a still smaller sized file. Examine the file images to see if they are of acceptable quality for screen reading.
5. If still smaller sizes are needed, set the compression setting to *screen* , compatibility to 1.5 and press the `Shrink PDF` button. This produces the smallest file size. If the images are acceptable - great but if not, keeping the preset setting to screen and compatibility setting to 1.5, try increasing the image resolution to 100 with say a downsample threshold of 1 or 2. This will result in a slightly larger file size but might improve the quality of images in the PDF. In this

manner, we can get a file size between the ebook and screen presets.

6. You can also try other compatibility settings like 1.7 or 2.0 and see if they result in better compression as compared to the default 1.4 or 1.5.
7. After compressing a PDF, always manually examine the quality of images, charts and graphs. If the quality is unacceptable, move to a higher resolution through advanced settings till you get acceptable results.
8. If possible, try opening your compressed PDF in some PDF readers like Adobe Acrobat, Okular, SumatraPDF, Foxit PDF Reader, Google PDF Viewer / Drive, Apple Books and some popular web browsers like Chrome, FireFox and Edge just to test that the PDF can be easily read in commonly available PDF readers.
9. Compatibility setting 1.3 flattens the PDF which means that elements like transparency, layers, fields etc are all merged into a single layer. This may increase the size of the file rather than compress it. Use this options if (a) you want to ensure compatibility with very old PDF readers or (b) if there is specific requirement to *flatten* PDFs.

8. Troubleshooting

While compressing a document, , Windows users might infrequently encounter the following error (Figure 7):

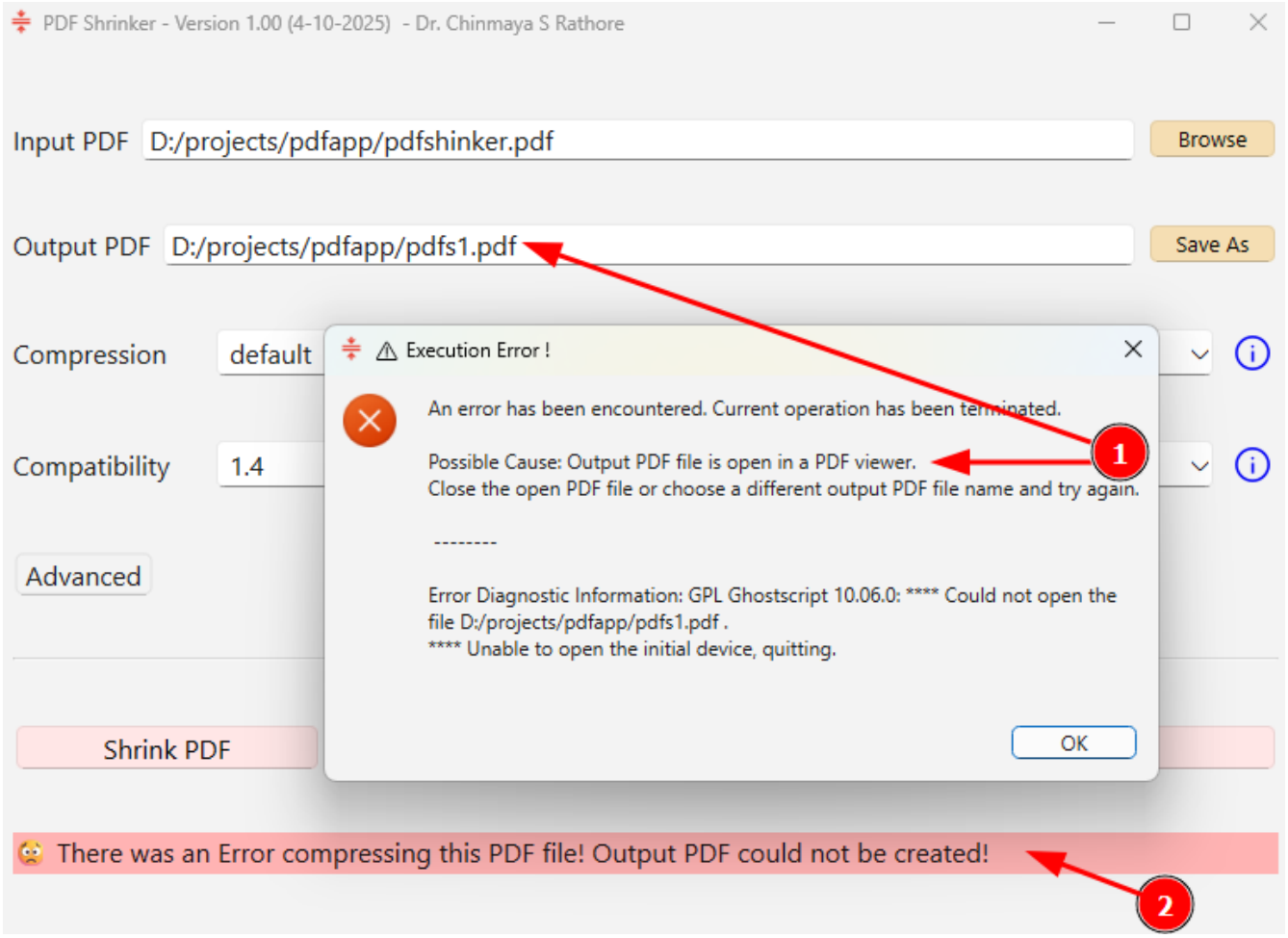

Figure 7. File open error

This happens when the file named in the output file name field is open in a PDF viewer (1). As the

file is already open and being viewed, PDF Shrinker cannot overwrite it. Hence an error is notified in the message box and also on the app message bar (2). To resolve this issue, just close the open file and press the `Shrink PDF` button again or choose a different file name for the output file and press the `Shrink PDF` button.

PDF Shrinker also does not permit input and output file names to be identical.

Appendix A: Checking the SHA256 Checksum

A SHA-256 checksum is a unique, fixed-length digital fingerprint generated from a file or message using the SHA-256 cryptographic hash function. It is used to verify the integrity and authenticity of data, ensuring that a file has not been corrupted or tampered with during download or transmission. If a file's SHA-256 checksum matches the original, it means the file is identical to the original version.

To match the SHA256 checksum value of the downloaded installer or package with the published checksum on the project page, do the following :

A.1. Windows

Open a terminal, and navigate to the folder where you have the downloaded pdfshrinker installer (pdfshrinker_1.0.0_x64-setup.exe) and type in the following command (Figure 8(1)):

```
certutil -hashfile pdfshrinker_1.0.0_x64-setup.exe SHA256
```

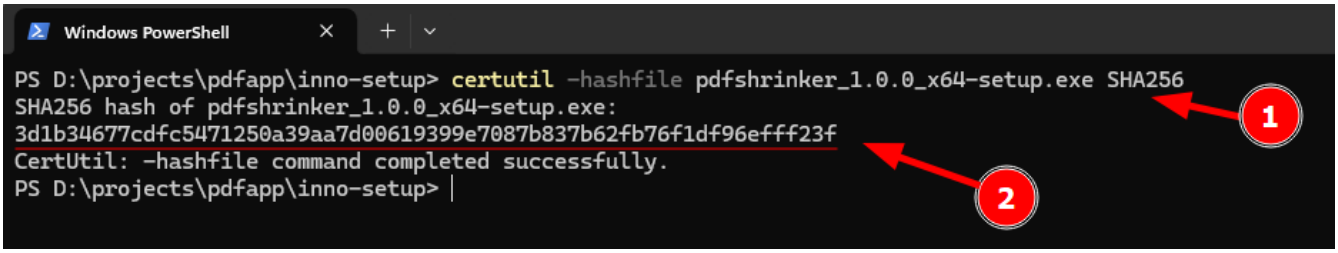

Figure 8. Calculating the SHA256 hash checksum of the downloaded package

You should see a long hash value as can be seen in Figure 8(2). Copy this value in notepad and paste the hash value published on the releases page and compare. They should be the same indicating that the file that you have downloaded has not been tampered with since it was uploaded by the developer from where you have downloaded it.

A.2. Linux

A.2.1. Ubuntu/Debain

Downloaded the deb-SHA256.txt and run the following command from a terminal *after navigating to the folder* that has the downloaded deb package:

```
sha256sum -c deb-sha256.txt
```

If you get a response that says : OK , the sums match !

A.2.2. REHL/Fedora

Downloaded the rpm-SHA256.txt and run the following command from a terminal *after navigating to the folder* that has the downloaded rpm package:

```
sha256sum -c rpm-sha256.txt
```

If the two checksums match, you should see an **OK** response as shown in Figure 9.

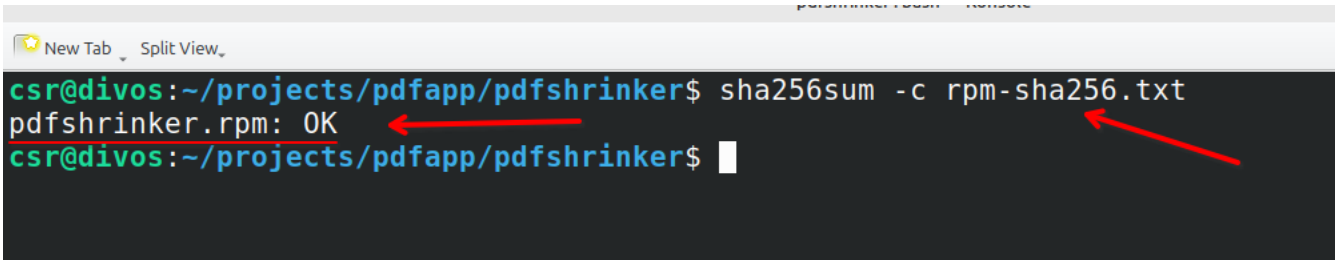

Figure 9. Comparing SHA256 checksums automatically on linux