



Datalogics PDF Optimizer™

User Guide

PDF Optimizer version 3.5

©2017-2024 Datalogics, Inc. All rights reserved.

Use of Datalogics software is subject to the applicable license agreement.

PDF Optimizer is a trademark of Datalogics, Inc.

For additional information, contact:

Datalogics, Incorporated

1207 Delaware Ave.

Suite # 1810

Wilmington, DE 19806

Phone: (312) 853-8200

www.datalogics.com

Table of Contents

| | |
|---|----|
| Introduction | 1 |
| What you get when you buy PDF Optimizer | 1 |
| How to use PDF Optimizer | 1 |
| The PDF Optimizer Profile..... | 2 |
| Installing PDF Optimizer: PDF Checker | 4 |
| Installing PDF Optimizer: Updating the Activation Key | 4 |
| Command Syntax..... | 5 |
| How PDF Optimizer processes a PDF document | 6 |
| Setting up your profile..... | 7 |
| Images..... | 9 |
| Downsampling Images..... | 9 |
| Downsampling and Recompression | 9 |
| Image Resolution | 10 |
| A Few Examples of JSON Profile Settings | 11 |
| Images: Color | 12 |
| Images: Grayscale | 13 |
| Images: Monochrome..... | 14 |
| Images: Color, Grayscale, or Monochrome | 15 |
| Fonts | 16 |
| Transparency | 17 |
| Objects..... | 18 |
| User data..... | 19 |
| Cleanup | 19 |
| General..... | 20 |
| Color Conversion..... | 20 |
| PDF/A Conversion | 24 |
| PDF/X Conversion | 25 |
| Windows Error Codes | 26 |
| Linux Error Codes..... | 27 |

Introduction

PDF Optimizer is a command line tool for Windows or Linux 64-bit platforms that you can use to optimize PDF documents. With this product you can count on maximum file size reduction and enterprise-level accuracy and reliability. PDF Optimizer offers flexible control, allowing you to optimize your documents for a wide range of document workflows. You can optimize a PDF document so that it will download faster, open more quickly in a web browser or PDF viewer, print more efficiently, or be better suited for long-term preservation. Select exactly what type of content you prefer to preserve or discard, fine-tune image compression options, and optionally convert to a PDF/A or PDF/X standard. With PDF Optimizer, you can ensure that your PDF documents will always be as small as possible while still meeting your needs.

Optimizing can also improve the performance of a PDF document without necessarily making that document smaller. The optimization process can discard objects and features within a PDF document that require excess processing time. A complicated PDF document with a lot of features will tend to take more time to load, regardless of the size.

We provide detailed online documentation about PDF Optimizer and our other products at our Developer Resources site. Visit <https://docs.datalogics.com/PDFOptimizer> to learn more.

What you get when you buy PDF Optimizer

- A Windows or Linux command line tool for making PDF documents smaller or more efficient, or both
- The capacity to create and use JSON profile files, each one providing a list of settings for processing PDF documents from a command line or as part of an automated workflow. JSON, or JavaScript Object Notation, is an open standard file format that relies on easily readable English text, and it is used as an alternative to XML.
- Technical support from our team of digital document specialists and professionals. You can contact your Datalogics Support representative directly by electronic mail or visit our support site for this product.
- Regular software updates
- Comprehensive user documentation, provided as a PDF file and online at our Developer Resource site.

How to use PDF Optimizer

You can run PDF Optimizer from a command prompt, and manually optimize one PDF document at a time. You can add a PDF Optimizer command to a batch or shell script file to automate the process. This way you can create a workflow that uses PDF Optimizer to optimize large numbers of PDF documents automatically. You can use your favorite scripting language to automatically generate a series of calls to PDF Optimizer, each with a unique name for the input PDF document and the output file. PDF Optimizer is a quick and easy way to convert a set of PDF files to make them easier to distribute.

When you run PDF Optimizer, the program takes the input PDF document you provide, optimizes it, and then saves it as a new output PDF document.

After PDF Optimizer finishes processing your document, you will see a summary statement describing how much the product has reduced the size of your source file, and what was changed or removed to make the output PDF document smaller. This summary appears by default, but if you don't need the summary you can disable it. You can also choose to generate an optional detailed Results Report, saved as a text file in a folder you select.

Both of these reports are designed to make PDF Optimizer easier for you to use. As soon as you use the software to optimize one of your PDF documents, the product can provide details for you about the results—how much space was saved by reducing the resolution of individual images in the original document, embedding fonts, and removing objects like thumbnails, annotations, and JavaScript code.

The PDF Optimizer Profile

You can provide a variety of custom settings to tell PDF Optimizer exactly how you want it to optimize each PDF document. For example, you can make a PDF smaller by choosing to downsample the images found in that document. Downsampling an image reduces the resolution of that image and reduces its size as well. You could add a setting to a profile file that tells PDF Optimizer to discard bookmarks or attachments within a PDF document.

The custom settings that you provide to PDF Optimizer to manage the process are all defined in a **profile** file. The profile is a JSON file that you need to specify in the command line statement.

We provide a set of profile files with the product, including several for converting documents into PDF/A output. These JSON files are stored in the OptimizationProfiles folder:

- `compressionMedium.json`. – This is the profile recommended for PDF documents to be optimized without a specific case in mind. It provides strong compression, and the loss in quality of images is minimal.
- `compressionHigh.json` – This profile is designed for maximum compression. For some documents this may mean a significant loss in the quality of images.
- `compressionLow.json` – This profile is designed for modest compression, and little to no loss in the quality of images.
- `Digimaster.json`. – This profile color converts images within a PDF document to gray scale and downsamples any images greater than 600 DPI resolution to 600 DPI. This profile is intended to be used with Kodak Digimaster printers.

- We provide a series of profiles designed to create an archived version of a document for long-term storage.
 - PDF/A-1b.json. – Basic conformance with PDF/A archival standards.
 - PDF/A-2b.json. – Basic conformance with archival standards but revised for later versions of the PDF format. PDF/A-2 includes options for OpenType fonts, layers, attachments (which themselves must be PDF/A compliant) and JPEG 2000 image compression.
 - PDF/A-2u.json. – Matches PDF/A-2b but also requires that all text in the document have Unicode mappings.
 - PDF/A-3b.json. – Matches PDF/A-2b, except that it is possible to embed any kind of file in the PDF document. With PDF/A-3 a user can save an XML, CSV, CAD, spreadsheet, or other type of file in the PDF document and still conform to PDF/A standards. The file embedded in the PDF/A-3 does not need to be converted to PDF/A itself.
 - PDF/A-3u.json. – Matches PDF/A-3b, but also requires that all text in the document have Unicode mappings.
 - PDF/A-4.json. – Based on PDF 2.0
 - PDF/A-4e.json. – Based on PDF 2.0 for Engineering documents
 - PDF/A-4f.json. – Based on PDF 2.0 for embedded files in any format
 - PDF/X-4.json. – Conformance with PDF/X-4 archival standard.
 - PDF/X-6.json. – Conformance with PDF/X-6 archival standard.
- printing.json. This profile is recommended for PDF documents to be sent to a printer.

You can update these files as you like, but the content of your profile file must be valid JSON content.

If you want to create your own custom profile, you can base your work on one of the profiles provided with the software installation package. But we recommend that you save a copy of the original JSON file and rename it, and then edit this copy instead. This way you will preserve the original JSON profile for later reference. Also, if you install an updated version of the software, the installation process will overwrite the original file, and any changes you made to that file will be lost. Besides saving your own copy of the profile, you might also want to create a backup of your edited JSON profile in a different directory.

You can use a JSON validator, like JSONLint, to check your JSON syntax (<https://jsonlint.com>).

Installing PDF Optimizer: PDF Checker

Note that when you install PDF Optimizer, you can also install a separate utility as well called PDF Checker. PDF Checker is a simple command line utility for 64-bit Windows and Linux platforms that allows you to quickly scan a PDF document or set of documents to look for problems, or to simply identify features within a document that are likely to get in your way if you want to use the PDF efficiently. Then, you can use this knowledge of your PDF documents to enhance them using PDF Optimizer. The two tools are designed to work together.

Installing PDF Optimizer: Updating the Activation Key

When you first purchase PDF Optimizer, you will receive an activation key, and you will be prompted to enter this key when you install the software. In response PDF Optimizer generates a license file for you (pdfoptimizer.lic) and stores it in the software installation directory.

When your evaluation period ends, and you later receive a new activation key for PDF Optimizer, you will need to enter it as you did your original activation key when you first installed the product. You will be prompted to enter the activation key. Type or paste the key value as you did before. PDF Optimizer will update the license file for you, and you can continue to use the product.

Note that if you don't enter the activation key value when you first install the product, or enter it incorrectly, you will be prompted to enter the value again the next time you run PDF Optimizer.

Command Syntax

The command syntax for PDF Optimizer must include:

- `pdfoptimizer` the executable file name
- `-i [--input]` name of the PDF Input file you want to optimize
- `-o [--output]` name you want to assign to the output PDF file
- `-j [--profile]` name of the JSON profile file

For each command line option, you can use the short ("`-i`") or long ("`--input`") notation.

These parameters are optional:

- `-q [--quiet]` don't display the results summary on the command line tool
- `-l [--log]` save detailed results to an output text file

A command statement for PDF Optimizer might look like this:

```
pdfoptimizer --input March_Report.pdf --output March_ReportB.pdf --  
profile compressionMedium.json
```

You don't need to include the path for any of these files if the input file and profile are stored in the same directory as the program executable, and if you save the output file to the same directory as well.

But you might want to draw an input file from one directory and save the output to another. In that event you need to provide the path as well as the file name. The examples that follow are for Windows:

```
pdfoptimizer --input C:\Datalogics\OptimizedFiles\AnnualReport2016.pdf  
--output C:\Datalogics\OutputFiles\AnnualReport2016-B.pdf --profile  
compressionMedium.json
```

If any of the file or path names include spaces, use quotes around the name:

```
pdfoptimizer --input "C:\Datalogics\Optimized Files\Annual Report  
2016.pdf" --output C:\Datalogics\OutputFiles\AnnualReport2016-B.pdf --  
profile C:\Datalogics\profiles\images.json
```

Note that PDF Optimizer allows for a command statement syntax that does not include the option notation, for compatibility with previous versions of the software. That is, you can leave out "`--input`" and "`--output`" if you already have this command syntax in place in your existing PDF Optimizer work flow. A command that simply lists the name and path of the input file, output file, and JSON profile might look like this:

```
pdfoptimizer MarchReport2016.pdf MarchReport2016-B.pdf  
compressionMedium.json
```


You will see a summary statement in your command line tool after you run the software to optimize a PDF document. If you would rather not see this statement, you can disable it with the optional "-q" (--quiet) parameter, like this:

```
pdfoptimizer --input March_Report.pdf --output March_ReportB.pdf --  
profile compressionMedium.json --quiet
```

On the other hand, you can add a parameter to generate an optional Results Report, exported to a text (.txt) file, using the "-l" (--log) parameter. Include name of the report file:

```
pdfoptimizer --input March_Report.pdf --output March_ReportB.pdf --  
profile compressionMedium.json --log MarchResults.txt
```

If you want to save the results report to a specific folder, include the path name for that folder in the command line statement:

```
pdfoptimizer --input March_Report.pdf --output March_ReportB.pdf --  
profile compressionMedium.json --log C:\DataLogics\Results  
Report\MarchResults.txt
```

Note that you can use both the --quiet and --log parameters in the same statement if you like. That is, you can suppress the summary report that appears on your command line and generate a detailed output report at the same time.

The Windows installation adds the location of the PDF Optimizer executable to the %PATH% in the Windows Environment Variables, so you can run "pdfoptimizer.exe" from anywhere. For Linux, you need to add the location of the PDF Optimizer executable to your PATH variable.

How PDF Optimizer processes a PDF document

When PDF Optimizer processes a PDF document, it follows a series of steps described below. You can define the settings involved in the optimization process, such as whether to remove embedded fonts and how much to compress images, in your JSON profile file. Also, several of the steps in the process are optional:

1. Flattening transparencies found in the PDF input document (optional)
2. Converting the colors in the PDF input document to a target color space, using a color profile and render intent provided by the user (optional)
3. Optimizing content, including images, fonts, objects, and user data, and then the objects are compressed as a part of the cleanup process.
4. Converting the optimized document so that it is compliant with the PDF/A or PDF/X specification (optional).

These steps are described in detail in this document.

Setting up your profile

Your JSON profile file should include a list of settings that define exactly what kinds of changes you want to apply to your PDF document. The PDF Optimizer offers a lot of different methods to optimize of a PDF document. The options you select will depend on how you want to change your output PDF documents.

Suppose you have a PDF document that is 18 MB and you want to make it smaller so that the file will be easier to distribute online. If you expect that your readers will be opening the file in a browser window, and it doesn't matter if the photographs and diagrams in the document appear with a lower resolution, you could make the document smaller by aggressively compressing the images included in the file.

On the other hand, if you are working with a large PDF document that your customers are likely to want to print, but you want to make it smaller so that it downloads and prints more quickly, you probably want to leave the graphics alone. They will need to appear as sharp as possible. But you don't need interactive content, like form fields, bookmarks, comments, or digital signatures. You can use PDF Optimizer to remove items from the PDF document that will not appear on paper.

Consider building a PDF document that you intend for people to read on smart phones and other mobile devices. In this case you want to compress the document so that it opens as quickly as possible. So you would reduce the size of the images in this case as well, but given that the screens are a lot smaller than a laptop or desktop monitor, you can reduce the resolution of the images in the PDF document to be less than a PDF document that is intended for opening in a browser window.

All of the settings for compressing a document are optional and are turned off by default. That means then that a setting is only applied if it is included in the JSON file. Flag settings must be set to "ON." Settings that are turned off do not need to be defined in the JSON profile file. So if you wanted to you could create a custom JSON file with only a single setting, to compress images. Your JSON file might only hold five or six lines of text.

Only use lowercase characters for the keys and values you add to the JSON file.

The methods you can use to optimize a PDF document are sorted into different categories:

| | |
|--------|--|
| Images | When a PDF document is created that includes photographs, diagrams or drawings, the original graphic file, such as a JPEG photograph or a PNG image, become images in that PDF file. You can enter settings in the JSON profile file to compress these color, gray scale, or black & white images. |
| Fonts | PDF documents can travel with the fonts that they need to access to properly render text. Font files can be embedded within the PDF itself. That way, no matter what machine is used to open a PDF file, the PDF is always guaranteed to look the same, and the viewing tool does not need to look for substitute fonts installed on the local desktop or laptop. But these embedded font files can make the PDF larger, maybe a lot larger, if the document needs to express characters |

from an Asian font set, like Mandarin or Japanese. You can enter settings in your JSON profile file to remove individual font characters or sets that you don't need, thus reducing the size of the PDF file.

Transparency

It is possible to stack objects, such as graphics, images, text boxes, and form fields, on top of each other on a PDF document. These objects can be partially or fully transparent, and thus can interact in various ways with objects behind them. If a set of transparencies are stacked in a PDF file, each one contributes to the final result that appears on the page, such as the colors blending together into a final color that appears.

To make a PDF document simpler you can flatten these transparencies. The flattening process combines the layers of content on a PDF page, or a stack of transparent images or colors, and renders the result as a single image, blended color, or set of text. For example, if a digital signature is flattened, the digital certificate key and related properties are removed from the signature field. The name of the person who signed the document and related information, such as the date and time stamp and the signer's email address, appear on the page as text, but the signature field is no longer interactive.

To flatten the transparencies within a PDF document, there needs to be a "transparency/quality" section in the JSON profile file.

Objects

Besides graphic images and font files, a variety of other objects can be saved within a PDF document.

- Blocks of JavaScript code
- Thumbnail images
- Bookmarks
- Tags
- Alternate graphics images

PDF Optimizer is designed to allow you to remove any of these objects from a PDF document. This serves to make the document smaller and easier to distribute.

User Data

It is possible to edit PDF documents using Adobe Acrobat and other viewing and editing tools. For example, when reviewing the content in a PDF document, a user might want to add a comment. It is also possible to attach external files to a PDF document so that the file is saved as a part of the PDF, or embed a hyperlink to a web page. Finally, a user could add metadata. PDF Optimizer can remove any of this content. It can also remove form fields, such as text boxes, check boxes, and radio buttons.

| | |
|------------------|---|
| Cleanup | Use the Cleanup features in PDF Optimizer to set compression values for a PDF document. You can compress the entire PDF document or parts of the content, and you can also remove redundant content or select a compression method to use, as well as other changes designed to make a PDF document open more quickly. |
| General | These options affect the PDF document as a whole, rather than individual features, like images or fonts or bookmarks. The setting provided under General allows you to force PDF Optimizer to generate a PDF output document even if this document will be larger than the input document it is intended to replace. |
| Color Conversion | You can turn on this feature to direct PDF Optimizer to convert the colors in a PDF document to a color profile you select before optimizing that document. |
| PDF/A Conversion | A PDF/A, or PDF Archive document, is a type of PDF file that is designed to be stored so that it can be accessed for many years to come. PDF/A documents must be able to be opened and read using viewing tools available in the future, so they are designed to be self-contained. For example, all of the fonts used in a PDF/A document must be embedded within the PDF document itself for the file to be considered PDF/A compliant. You can turn on this option and prompt PDF Optimizer to save a PDF file as PDF/A compliant after the document is optimized. You can also define the color space used and determine how PDF Optimizer respond to errors during the conversion process. |
| PDF/X Conversion | A PDF/X, is an ISO-standard version the PDF format. It's a subset of the PDF standard that applies to Printing Workflows. |

Images

Downsampling Images

If you have images in a PDF document that you want to make smaller, and you know that these images don't need to have a high resolution in the output file, you can reduce the resolution of these images. You can also compress these images within the file. Both steps will reduce the final size of the PDF document.

In PDF Optimizer, the process of reducing the resolution of images is called downsampling. You can choose to downsample color images in a PDF document, or grayscale, or monochrome (black & white). The settings for reducing the resolution for these three kinds of images in a PDF document must be added separately to the JSON profile file. Each type of image can have its own settings and resolution values. So you could, for example, enable resampling to only apply to the color images in a PDF document. Or you could include only grayscale and black and white images.

Downsampling and Recompression

Downsampling reduces the size of the image directly by reducing the resolution. In recompression, compressed images in a document are decompressed and then compressed again. You can enter a

recompression setting to change the compression algorithm used for recompression, such as ZIP, JPEG or Flate, and another setting to change the final image quality. The image quality is part of the compression method used. If you add settings in the JSON profile file to downsample images, PDF Optimizer will also recompress the images involved whether you provide recompression settings or not.

If you do not add recompression settings to the JSON profile, PDF Optimizer downsamples and recompresses each image in the PDF document using the default compression algorithm and quality value defined in the image itself. For example, if you provide downsample settings but not recompression settings in your JSON profile and apply that profile to a document that only holds images using JPEG compression, PDF Optimizer will use the JPEG compression method. It will also use the highest quality recompression setting available (“maximum”) to keep from reducing the quality of the images as they are recompressed.

On the other hand, if you decide to leave out downsample settings from your JSON profile file, but add recompression settings, PDF Optimizer recompresses the images using the recompression algorithm you provide while keeping the image downsampling resolution (DPI) the same. Note that if you add recompression settings you must include both values in the JSON file, the compression algorithm and the recompression quality level.

Image Resolution

When we refer to the resolution of an image, we generally refer to the number of pixels in that image. This can be expressed in terms of megapixels, or in Dots per Inch (DPI). With an image in a PDF document, the resolution of the image is expressed as a certain number of pixels wide and pixels high. The downsampling process involves changing the width and height of an image in pixels, in order to reach a given target resolution. PDF Optimizer calculates the resolution for every image in the document. Keep in mind that the resolution values used with downsampling are distinct from the image quality settings used for image recompression.

You can specify a target resolution to use for downsampling images in a document (target-dpi) and a trigger resolution (trigger-dpi). If you decide to downsample an image type, both the target and the trigger resolution settings must be included in your profile file. The target resolution defines the goal—the maximum resolution for every image in the file. So if you add a target resolution to your JSON profile and set that target resolution to 600 DPI, PDF Optimizer will downsample every graphic in the PDF document to 600 DPI unless it that image is already at 600 DPI or less.

The trigger resolution, if used, defines the resolution PDF Optimizer uses as its starting point. Any image with a resolution greater than the trigger resolution will be downsampled. If an image has a resolution less than the trigger resolution, PDF Optimizer ignores it. So if you set the trigger resolution to 800 DPI, and the target resolution to 400 DPI, it means that you want to downsample every image in the PDF document to 400 DPI, but only if the image is larger than 800 DPI to begin with. In this example you would be telling PDF Optimizer to look for only the really large images (the ones with a resolution at 800 DPI or more) and then downsample just those images to a certain set value, in this example 400 DPI.

If the trigger resolution is 500 DPI, and the target resolution is 400 DPI, PDF Optimizer will not downsample an image if it is 480 DPI. But if the trigger resolution is 500 and the target is 400, if PDF Optimizer finds an image with a resolution of 680 DPI, it will downsample it to 400 DPI.

A Few Examples of JSON Profile Settings

This example shows settings used to downsample and recompress color JPEG images in a PDF document:

```
"color": {
  "downsample": {
    "trigger-dpi": 225,
    "target-dpi": 150
  },

  "recompress": {
    "type": "jpeg",
    "quality": "medium"
  }
},
```

Here images with a resolution above 225 DPI will be resized to 150 DPI, using the JPEG compression algorithm, and set at "medium" quality. The quality levels (low, medium, high) are defined within the JPEG algorithm itself.

In this example, we don't provide recompress settings.

```
"color": {
  "downsample": {
    "trigger-dpi": 225,
    "target-dpi": 150
  },
},
```

PDF Optimizer will use the same compression algorithm, JPEG, because these types of images are by default JPEG compressed. But the quality setting will be "Maximum" because the software tries to maintain the same level of compression quality as found in the original JPEG images.

Finally, consider this example:

```
"color": {
  "downsample": {
    "trigger-dpi": 225,
    "target-dpi": 150
  },
  "recompress": {
```

```

        "type": "zip"
        "quality": "medium"
    },
    },

```

This refers to the same type of image in the PDF document, color JPEG images. PDF Optimizer will downsample the color images to 150 DPI if they are above 225 DPI, but uses the ZIP compression algorithm, rather than using the default JPEG compression method.

Images: Color

| | | | |
|------------|--|--|---|
| downsample | Ability to specify a target resolution and a trigger resolution at which color images will be recompressed. | | |
| | trigger-dpi | All color images above this resolution will be downsampled. | |
| | target-dpi | The new resolution of downsampled color images. | |
| recompress | Sets the type and quality of compression used to downsample color images. JPEG compression is a compression format commonly used for photographs. It uses a technique known as DCT, Discrete Cosine Transform. | | |
| | type | same | Keep original default compression algorithm provided in the images themselves |
| | | zip | Use ZIP compression |
| | | jpeg | Use JPEG compression |
| | | jpeg2000 | Use JPEG2000 compression |
| | | zip-jpeg | Use both ZIP and JPEG compression for the images |
| | quality | These values are valid for JPEG and JPEG2000 compression only: | |
| | | minimum | |
| | | low | |
| | | medium | |
| | | high | |
| | | maximum | |
| | | lossless | Original quality of the graphic is preserved (see Monochrome). Only available for JPEG2000. |

Images: Grayscale

| | | | |
|------------|--|---|---|
| downsample | Ability to specify a target resolution and a trigger resolution at which grayscale images will be recompressed. | | |
| | trigger-dpi | All grayscale images above this resolution will be downsampled. | |
| | target-dpi | The new resolution of downsampled grayscale images. | |
| recompress | Sets the type and quality of compression used to downsample grayscale images. JPEG compression is a compression format commonly used for photographs. It uses a technique called DCT, Discrete Cosine Transform. | | |
| | type | same | Keep original default compression algorithm provided in the images themselves |
| | | zip | Use ZIP compression |
| | | jpeg | Use JPEG compression |
| | | jpeg2000 | Use JPEG2000 compression |
| | | zip-jpeg | Use both ZIP and JPEG compression for the images |
| | quality | These values are valid for JPEG and JPEG2000 compression only: | |
| | | minimum | |
| | | low | |
| | | medium | |
| | | high | |
| | | maximum | |
| | | lossless | Original quality of the graphic is preserved (see Monochrome). Only available for JPEG2000. |

Images: Monochrome

| | | | |
|------------|---|--|---|
| downsample | Ability to specify a target resolution and a trigger resolution at which monochrome images will be recompressed. | | |
| | trigger-dpi | All monochrome images above this resolution will be downsampled. | |
| | target-dpi | The new resolution of resampled monochrome images. | |
| recompress | <p>Sets the type and quality of compression used to downsample monochrome images.</p> <p>JBIG2 is a compression algorithm designed for binary images, or images where each pixel can only have one of two possible colors. It can be used for either lossy or lossless image processing.</p> <p>CCITT Group 4 refers to the compression type from the International Telegraph and Telephone Consultative Committee (CCITT) or TIU. Many fax and document imaging file formats support this form of lossless data compression encoding. These protocols are referred to as CCITT Group 3 and Group 4 compression, respectively.</p> <p>Lossy and lossless refer to the approach used for compressing data. For lossless, all of the data in the image is preserved. The quality of the image does not change, and it can be uncompressed to its original state. Lossy compression permanently removes data from the image file, such as pixels, reducing the image resolution. Files reduced using lossy compression will be considerably smaller, but will not print or display as well as those compressed using lossless compression.</p> | | |
| | type | same | Keep original default compression algorithm provided in the images themselves |
| | | jbig2 | Use jbig2 compression |
| | | ccittg3 | Use ccittg3 compression |
| | | ccittg4 | Use ccittg4 compression |
| | quality | lossy | Valid for jbig2 only |
| | | lossless | Valid for jbig2 only |
| | | | |

Images: Color, Grayscale, or Monochrome

Optional. Enable any of these values by adding it to the profile and setting it equal to ON.

| | |
|---|--|
| optimize-images-only-if-reduction-in-size | Set this value to ON if you want PDF Optimizer to only downsample an image found in a PDF document if the newly downsampled image is in fact smaller than the original. When recompressing an image, the output file can possibly expand in size. If the process yields an image that is the same size as the original, or larger, PDF Optimizer will leave the image alone. |
| consolidate-duplicate-image-and-forms | Remove duplicate copies of alternate images and forms. |
| down-convert-16-to-8-bpc-images | <p>When enabled, images that are 16 bits per component will be converted to 8 bits per component.</p> <p>The color depth of an image is the number of bits used per pixel for each color component. RGB, for example, has three color components. By down-converting an image in a PDF file from 16 bpc to 8 bpc, you are reducing the resolution of the image, but also significantly reducing its size. If a PDF document features high-resolution images, the final PDF can also be significantly smaller.</p> <p>This feature is not applicable if</p> <p>Color Conversion is enabled (see page 20). Color Conversion will attempt to down-convert 16 bpc images automatically if you turn it on. The down-convert feature only has an impact if Color Conversion is turned off.</p> |

Fonts

Optional. Enable any of these values by adding it to the profile and setting it equal to ON.

| | |
|-----------------------------|---|
| subset-embedded-fonts | <p>Subsetting fonts removes unused characters from font files embedded in the PDF.</p> <p>It is a best practice when working with PDF documents to embed every font used in that document into the document itself. That way, the viewing tool (like Acrobat) does not have to look for a font on the local system, or choose a substitute. But embedding font files in a PDF document can make the PDF quite large, especially if the PDF has embedded a font file for an Asian language, such as Mandarin, with tens of thousands of glyphs.</p> <p>A glyph is a symbol that represents a character in a font. For example, Times Roman has its own a glyph for the letter “a,” as does Helvetica, Courier, and Tempus sans. The single letter, lower case a, could be expressed in a PDF file as any one of hundreds of different kinds of glyphs.</p> <p>To avoid this, a subset of the glyphs in the font can be saved in the PDF document. The subset font only includes the characters you expect to need when rendering the pages of that document. This often leads to a much smaller PDF.</p> |
| consolidate-duplicate-fonts | <p>Remove multiple copies of the same font file. Sometimes PDF documents are created with multiple copies of the same font, either as multiple subsets or multiple, fully embedded copies of a font file. When multiple copies of the same font appear, they may be merged into a single font.</p> |
| unembed-standard-14-fonts | <p>Remove the Base 14 fonts. The Base 14 fonts are:</p> <ul style="list-style-type: none">• Times Roman• Helvetica• Courier• Symbol• Times Roman bold• Helvetica bold• Courier bold• ZapfDingbats• Times Roman italic• Helvetica oblique• Courier oblique• Times Roman bold/italic• Helvetica bold/oblique• Courier bold/oblique <p>When these standard fonts are embedded in a PDF document they can make the file larger.</p> |

| | |
|-----------------------|---|
| resubset-subset-fonts | <p>Allows already subset fonts to be re-subset if possible (see “subset-embedded-fonts” above). Subsetting can significantly reduce the size of the PDF document if a font features thousands of glyphs, such as Mandarin. By re-subsetting a subset font in a PDF document, you are replacing it with a subset that will only contain the glyphs currently in use in the document.</p> <p>Suppose you have a long PDF document that uses Mandarin characters. You decide to create a summary version of this file by deleting all but the first two pages. After re-subsetting, the Mandarin characters that no longer appear in the document are removed.</p> |
| remove-unused-fonts | Removes fonts that are embedded but not being used in the document. |

Transparency

To flatten transparencies in a PDF input document, add one of these "quality" settings to your JSON profile file. If you don't include the "quality" setting in your JSON profile, PDF Optimizer will not flatten transparencies.

Note that the printing.json JSON profile included with PDF Optimizer provides the "transparency" section at the end of the file.

| | | |
|---------|--|---|
| quality | The resolution level to use when flattening transparent objects. The higher the level of quality, the better the final output in print or in a browser window. But the resulting PDF document will also be larger. | |
| | low-quality | Line art and text 288 DPI, gradients 144 DPI |
| | medium-quality | Line art and text 300 DPI, gradients 150 DPI |
| | high-quality | Line art and text 1200 DPI, gradients 300 DPI |

Objects

Optional. Enable any of these values by adding it to the profile and setting it equal to ON.

| | |
|----------------------------|--|
| discard-javascript-actions | Removes JavaScript content. Blocks of JavaScript code can be added to a PDF to complete a function or calculate a value, such as a user's age when that person enters his or her birth date in a form field. |
| discard-alternate-images | Removes alternate versions of the same image found in the PDF document. A PDF document can be set up to specify alternate images, or multiple versions of one image within the same document. These images can be used to meet different needs. For example, a PDF could present one image with a lower resolution for display on a monitor, and an alternate image with a higher resolution to use when the PDF document is sent to a printer. These images can make a PDF document very large. Today alternate images are rarely used. |
| discard-thumbnails | Removes document thumbnails. Thumbnail images are used to preview pages in a PDF document and appear in a panel on the left side of the viewer window. A user could scroll through a series of thumbnails to find a page he or she is looking for. |
| discard-document-tags | Removes document tags. Tags are sometimes added to PDF documents to provide structural information to describe items such as headers and other content on a page. Tagging is generally used with a PDF document to meet accessibility requirements. For example, tags in a PDF document might be placed so that text, headings, footnotes, and other content in the document can be interpreted by a screen reading software tool. |
| discard-bookmarks | Removes document bookmarks. Bookmarks make navigation easier. They are typically presented as a Table of Contents, and are commonly attached to headings within the document. A user typically interacts with this Table of Contents to move directly to the part of the page where the bookmark is found. And bookmarks can be used apart from a table of contents to mark a place in a PDF document to navigate to. |
| discard-output-intent | <p>Removes the document's output intent. The output intent is an entry in the PDF document OutputIntents array. The output intent is used to describe how the destination device for the document, most likely a printer, reproduces the colors in the document. Specifically, the output intent describes the ICC Color space to use for rendering the document. The ICC profile is stored in the PDF document itself. If the output intent is present, rendering will be to that profile. PDF documents that are compliant with an ISO standard, like Graphics Exchange (PDF/X), Archive (PDF/A), and Engineering (PDF/E), often have an output intent set.</p> <p>As the ICC profile can be quite large, you might want to remove it to reduce the size of the PDF document, if you don't have plans to print the document in the future. But if it is important to preserve color fidelity for your application, it should not be removed.</p> |

User data

Optional. Enable any of these values by adding it to the profile and setting it equal to ON.

| | |
|---|---|
| discard-comments-forms-multimedia | Removes interactive elements from the PDF document. These can include annotations, such as pop-up notes, comments, and highlights, as well as interactive Acrobat form fields and embedded multimedia. Besides removing the interactive elements, this option also removes the visual content associated with these elements. |
| discard-xmp-metadata-padding | XMP refers to the Extensible Metadata Platform, a standard created by Adobe Systems to guide the creation, processing, and exchange of metadata for a variety of digital resources. When XMP metadata is included in a PDF document, the application that creates the PDF leaves a “padding area” in the text stream, commonly 2 to 4 KB for each set of XMP metadata. This allows for the metadata to be edited in place, and expanded if needed, without disturbing the document as a whole. This option removes the XMP padding from the document. |
| discard-document-information-and-metadata | Removes document descriptions and metadata. A minimal set of information will be maintained to ensure maximum compatibility with PDF viewers and processors. |
| discard-file-attachments | Removes files attached to the document |
| discard-private-data | Removes piece data relevant to the application that created the file. Some applications, like Adobe Illustrator, add their own unique content to a PDF document when generating that document. This information is useful to the original software product if the PDF is opened and edited in that product again. But this information can be removed. |
| discard-hidden-layer-content | Layers in a PDF document allow content to be placed above or below other content. The content in layers can be hidden from view, so that, for example, copyright information provided on the bottom of a page does not appear in Adobe Reader but is included if the document sent to a printer. The layers in a PDF document can be displayed in the Layers panel in Adobe Acrobat. You can remove this hidden content to reduce the size of a PDF document. |

Cleanup

| | | |
|-------------|--|---------------------------------------|
| compression | Selects the compression action for the file. | |
| | compress-entire-file | Compress document as a single unit |
| | compress-document-structure | Compress the document structure only |
| | remove-compression | Removes compression from file streams |

Optional. Enable any of these values by adding it to the profile and setting it equal to ON.

| | |
|-----------------------------------|--|
| flate-encode-uncompressed-streams | Compress uncompressed streams using flate. |
|-----------------------------------|--|

| | |
|----------------------------|--|
| convert-lzw-to-flate | Recompress LZW-compressed streams using flate. |
| optimize-page-content | Removes redundant content streams, or page text. |
| optimize-for-fast-web-view | Place all the information needed to render the first page of the document near the beginning of the file. This is also known as linearization. |

Flate, or Deflate compression, is an open source standard widely used for creating zip files and in PDF documents. It is commonly used for PNG image files, and is much more widely used than LZW.

LZW, Lempel-Ziv-Welch, is a universal data compression algorithm, once widely used with Unix platforms. This method appears in some old PDF documents but it is rarely used any longer.

General

Optional. Enable either of these values by adding it to the profile.

| | |
|---------------------------------------|---|
| preserve-version | <p>If this value is set to off, PDF Optimizer saves the output PDF document to PDF version 1.7.</p> <p>If you set this value to “on,” PDF Optimizer preserves the original version of the PDF source document if possible. If you select an option for PDF Optimizer to use when optimizing your source PDF document that will conflict with the current version of the PDF document, PDF Optimizer will ignore your request to preserve the original version of the PDF source document.</p> <p>For example, if you want to use PDF Optimizer to process a PDF source document of version 1.5, and request that the version be preserved and also that the document be converted to PDF/A-3u format, the output will be 1.7 since it's required for PDF/A-3u output.</p> |
| write-output-even-if-increase-in size | <p>If this value is set to off, if PDF Optimizer finds that the output file will actually be bigger than the original PDF input file when it is finished processing, PDF Optimizer will not generate an output file at all. Rather, it will write an error message saying that the file could not be optimized.</p> <p>If you set this value to “on,” PDF Optimizer generates the PDF output file even if the optimization process produces a document bigger than the input file it is meant to replace.</p> |

Color Conversion

Optional. Enable this value by adding it to the profile and setting it equal to ON.

We can see thousands of different shades of colors, and high-quality digital cameras and scanners can often detect millions of shades. To manage the broad range of colors for producing graphics images in digital content, imaging professionals have developed models to define these colors, called color spaces.

Many color spaces have been defined. Some are dependent on hardware devices, and define what a camera can detect, or a printer print, or a monitor display. Others are based on software and thus can be used across many different types of devices, such as Adobe RGB or sRGB (standard RGB). A color space must be defined for any device or software product to make sure that coloring patterns remain the same from one device or system to another.

The Standard RGB color space, sRGB, was developed by Microsoft and Hewlett Packard to describe colors available on most monitors and displays. This color space is also commonly used for web graphics.

Adobe Systems' own Adobe RGB (Red/Green/Blue) color space is designed to hold all of the colors that are likely to be available on any color CMYK (Cyan/Magenta/Yellow/Black) printer. It is considerably larger than Standard RGB.

Color profiles are standards for managing colors, used to ensure that the colors for text or graphics in a file remain the same regardless of the hardware or software used to display, edit, or print that file. Color profiles are based on the specification created by the International Color Consortium (ICC) in 1993 to govern color and color management across all operating systems, platforms, and software and hardware and software systems. A color profile is usually expressed as a file included in the software or driver for an installed printer, scanner, or other hardware device, or in software used to edit a file that is to be displayed or printed. A profile provides a set of data that describes an input or output device. A color profile file can also be embedded in a PDF document.

| | | |
|-----------------|---|--|
| enabled | Enables PDF Optimizer to convert colors in a PDF document using a target profile. | |
| convert-profile | The profile used to convert colors in the PDF document. | |
| | lab-d50 | L*a*b color specification with a D50 white point. The Lab color space is based on the CIE XYZ color space, but it includes a dimension L, for lightness, along with a and b coordinates, to define the color. This is Adobe Systems' standard Lab profile. |
| | srgb | Standard RGB, the default profile for Windows monitors. |
| | apple-rgb | Apple RGB, the default profile for Mac monitors |
| | color-match-rgb | Color Match RGB. This is a simpler version of the Radius ColorMatch RGB space, without the non-zero black point. |
| | gamma-18 | Gray Gamma 1.8, grayscale display profile, used for content viewed on a monitor |
| | gamma-22 | Gray Gamma 2.2 |

| | | |
|----------------------|--|--|
| | dot-gain-10 | Grayscale printer profile, with dot gain 10%. Dot gain is commonly used in offset printing to define the increase in size in halftone dots in the printing process, making a printed document look darker than intended. |
| | dot-gain-15 | Dot gain 15% |
| | dot-gain-20 | Dot gain 20% |
| | dot-gain-25 | Dot gain 25% |
| | dot-gain-30 | Dot gain 30% |
| | monitor-rgb | RGB Monitor, referring to a monitor that requires separate signals for the three primary colors. |
| | acrobat5-cmyk | Adobe Reader 5 CMYK |
| | acrobat9-cmyk | Adobe Reader 9 CMYK |
| | <p>The above list of options represents the standard list of color profiles provided with PDF Optimizer. These color profiles are in the form of .icc profile files, and these files are stored in this directory after the product is installed:</p> <pre>PDF Optimizer\Resources\Color\Profiles</pre> <p>If you enter one of these options, PDF Optimizer can find the correct color profile file.</p> <p>If you want to use your own custom color profile file, provide the full path name where the file is stored.</p> <p>We recommend that you put your color profile file in the same \Resources directory where the other color profile files are stored, but you can store it wherever you like.</p> <p>If you want to apply your own color profile, the convert-profile section of the JSON profile file might look like this:</p> <pre> "color-conversion": { "enabled": "on", "color-convert-action": "convert", "convert-intent": "profile-intent", "convert-profile": "C:\DataLogics\PDF Optimizer\Resources\Color\ Profiles\custom_profile.icc" }, </pre> | |
| color-convert-action | <p>The type of color conversion. The values available include convert and decalibrate. The decalibrate setting means that PDF Optimizer will decalibrate calibrated color spaces in the PDF document.</p> <p>If a color profile file is assigned to a PDF document, or to a feature in of that PDF document, such as an image, the document or image is referred to as “calibrated.” This color profile will be used when rendering colors for that document.</p> | |

| | | |
|----------------|--|---|
| | <p>If a color profile is not assigned to the document, a default color profile is used instead, usually installed on the local device or on a printer. So, if you select decalibrate as the color-conversion-option, you can remove the ICC profile file stored in the PDF document, and thus reduce the size of the PDF output file.</p> | |
| convert-intent | <p>Use this setting to specify the color translation method for colors that are outside the gamut of the color profile. The intent lets the software determine how to substitute a color that can be written to the file. You can select from a list of standard strategies to apply when converting the colors in that original PDF document.</p> <p>The conversion intent is used to describe how the destination device for the document reproduces the colors in the document. Thus, when you print or display the PDF output file, the colors in the output file will match as closely as possible the original color found in the source PDF document.</p> | |
| | perceptual-intent | <p>Generally used for photography. This method does not map colors one for one but estimates to match colors. Hence it often provides the most pleasing result but not necessarily the most accurate.</p> |
| | relative-colorimetric-intent | <p>Generally used for photography. The relative method uses an algorithm to select the closest possible color map to be true to the specified color.</p> |
| | saturation-intent | <p>Commonly used in charts and diagrams with a limited palette of colors where hue is not as important.</p> |
| | absolute-colorimetric-intent | <p>Often used to select a specific color or set of colors for drawings or designs. Absolute serves to reproduce the exact colors provided in the original PDF document. A common reason for using absolute would be to reproduce the color used in a corporate logo such as IBM Blue. The color is changed by selecting a defined match. This method does not use a conversion algorithm to select the closest color available.</p> |

| | | |
|-------------|--|---|
| | profile-intent | If you specify a color profile in the convert-profile option the intent value defaults to that profile. In that case software will use the rendering intent provided with the ICC color profile currently in use for that PDF document. For example, the Adobe RGB 1998 color profile uses Relative Colormetric as its rendering intent. So if you specify Adobe RGB 1998 (srgb) as the color profile, and profile-intent is selected here, the PDF Optimizer software will use relative. |
| object-type | Use this setting to specify what type of objects are included in the color conversion process. This allows for more granular control over what types of objects are included or not during conversion. | |
| | all | Include all object types (images, text, line-art, and shading) during color conversion. |
| | images | Include images during color conversion. |
| | text | Include text during color conversion. |
| | line-art | Include line-art during color conversion. |
| | shading | Include smooth shadings during color conversion. |

PDF/A Conversion

Optional. Enable this category by adding it to the profile and setting “enabled” equal to on. This setting directs PDF Optimizer to save an optimized PDF document so that it is compliant with the PDF/A format. Other settings in this category allow you to control the conversion process and select the output type.

PDF/A, or PDF Archive, is an ISO-standard version of the PDF format. Adobe introduced PDF in 1993, and the International Organization for Standardization (ISO) took management of PDF as an open standard in 2008. ISO released PDF/A in 2005. PDF/A is designed to be used with PDF files that need to be archived and stored, and still capable of being opened and read in a viewing tool many years into the future. Hence the standards for PDF/A files differ from regular PDF documents. In one example, to be PDF/A compliant, the fonts in a PDF/A file must be embedded in the file itself so that a viewer does not need to try to access fonts stored in a font directory on a local machine.

| | | |
|---------|---|--|
| enabled | Enables conversion of the optimized PDF document to the PDF/A format. | |
| type | Select the type of PDF/A output to use for the target file: | |
| | 1b | Basic conformance with PDF/A archival standards |
| | 2b | Basic conformance with archival standards but revised for later versions of the PDF format. PDF/A-2 includes options for OpenType fonts, layers, attachments (which themselves must be PDF/A compliant) and JPEG 2000 image compression. |
| | 2u | Matches PDF/A-2b but also requires that all text in the document have Unicode mapping. |

| | | |
|---------------------------------|--|---|
| | 3b | Matches PDF/A-2b, except that it is possible to embed any kind of file in the PDF document. With PDF/A-3 a user can save an XML, CSV, CAD, spreadsheet, or other type of file in the PDF document and still conform to PDF/A standards. The file embedded in the PDF/A-3 does not need to be converted to PDF/A itself. |
| | 3u | Matches PDF/A-3b, but also requires that all text in the document have Unicode mapping. |
| | 4 | Based on PDF 2.0 |
| | 4e | Based on PDF 2.0 for Engineering documents. |
| | 4f | Based on PDF 2.0 for embedded files in any format. |
| pdfa-target-color-space | Select the target color space to use when converting the colors in the PDF document when converting it to PDF/A. | |
| | rgb | red green blue |
| | cmyk | cyan magenta yellow black |
| Rasterize-if-errors-encountered | <p>When set to OFF, if PDF Optimizer finds errors when saving the optimized PDF document as PDF/A compliant, PDF Optimizer ends the conversion to PDF/A and displays error message.</p> <p>When set to ON, if PDF Optimizer finds errors when converting a PDF document, it will rasterize the page with the problem into a graphic image and continue to save the document as a PDF/A document.</p> | |

PDF/X Conversion

Optional. Enable this category by adding it to the profile and setting “enabled” equal to on. This setting directs PDF Optimizer to save an optimized PDF document so that it is compliant with the PDF/X format. Other settings in this category allow you to select the output type.

PDF/X, is an ISO-standard version the PDF format. It’s a subset of the PDF standard that applies to Printing Workflows.

| | | |
|---------|---|---------------------------|
| enabled | Enables conversion of the optimized PDF document to the PDF/X format. | |
| type | Select the type of PDF/X output to use for the target file: | |
| | 6 | Conformance with PDF/X-6 |
| | 4 | Conformance with PDF/X-4 |
| | 3 | Conformance with PDF/X-3 |
| | 1a | Conformance with PDF/X-1a |

Windows Error Codes

- 1001 Syntax error. You started PDF Optimizer but left out the name of your input PDF document, or the export document, or the profile, or otherwise made an error in your command syntax.
- 1002 Profile not found. PDF Optimizer could not find the profile file you named in your command statement. Check the file name or path for errors.
- 1003 Profile invalid. The JSON file you provided as the profile contains syntax errors.
- 1004 Invalid flag value. PDF Optimizer found an unexpected value in the settings in your JSON profile. Key values, such as “medium” or “jpeg” as well as the word “on” must always be in quotes, while numeric values must not have quotes. This format would generate an error:
- ```
"target-dpi": "350"
```
- This is the correct format.
- ```
"target-dpi": 350
```
- Or if you leave out quotes, as if you meant to include a numeric value, that will also generate an error, like this:
- ```
"discard-javascript-actions": on
```
- This is the correct format.
- ```
"discard-javascript-actions": "on"
```
- 1005 Input PDF document was not found.
- 1006 PDF Optimizer could not create output PDF document. Verify that you have write permissions to the folder, that the file is not already open, and that the output file path name is valid.
- 1007 Input PDF document is signed or has usage rights and cannot be processed.
- 1008 Unable to write to log file.
- 1009 Syntax error. You did not include the name of your input file in your command statement.
- 1010 Syntax error. You did not include the name of your profile file in your command statement.
- 1011 Syntax error. You did not provide a name for creating an output file in your command statement.
- 1012 General processing error in creating output PDF document.
- 1013 The input file path is the same as the output file path.

Linux Error Codes

- 101 Syntax error. You started PDF Optimizer but left out the name of your input PDF document, or the export document, or the profile, or otherwise made an error in your command syntax.
- 102 Profile not found. PDF Optimizer could not find the profile file you named in your command statement. Check the file name or path for errors.
- 103 Profile invalid. The JSON file you provided as the profile contains syntax errors.
- 104 Invalid flag value. PDF Optimizer found an unexpected value in the settings in your JSON profile. Key values, such as “medium” or “jpeg” as well as the word “on” must always be in quotes, while numeric values must not have quotes. This format would generate an error:
- ```
"target-dpi": "350"
```
- This is the correct format.
- ```
"target-dpi": 350
```
- Or if you leave out quotes, as if you meant to include a numeric value, that will also generate an error, like this:
- ```
"discard-javascript-actions": on
```
- This is the correct format.
- ```
"discard-javascript-actions": "on"
```
- 105 Input PDF document was not found.
- 106 PDF Optimizer could not create output PDF document. Verify that you have write permissions to the folder, that the file is not already open, and that the output file path name is valid.
- 107 Input PDF document is signed or has usage rights and cannot be processed.
- 108 Unable to write to log file.
- 109 Syntax error. You did not include the name of your input file in your command statement.
- 110 Syntax error. You did not include the name of your profile file in your command statement.
- 111 Syntax error. You did not provide a name for creating an output file in your command statement.
- 112 General processing error in creating output PDF document.
- 113 The input file path is the same as the output file path.