

Optimization of existing PDF files: issues, constraints, and methods



YOUR DOCUMENT IMAGING PARTNER

This paper was originally a presentation in French delivered at the [PDF Day – France](#) by Loïc Carrère, CEO of ORPALIS, in April 2018.

Organized by the [PDF Association](#), the PDF days are the meeting place of the PDF industry, where experts conduct educational (non-commercial) presentations, panel and discussion-based sessions about the format.

Abstract: The richness of PDF offers many opportunities to reduce the weight of existing documents. Organizations which need to meet more and more legal requirements for archiving and data retention often adopt a strategy to reduce the amount of storage used by their existing documents. This paper will address the issues and constraints of such an approach as well as various optimization methods which can be applied. We will try to describe a maximum of optimization techniques, with or without loss of data, which can be adapted according to the expectations. We will discuss them with case studies dealing with documents of different nature (documents with vector content and documents containing only images).

ORPALIS at a glance

- More than fifteen years of experience in the graphic chain, from acquisition to archiving.
- Team of 25 people in France, Ukraine, Romania, and Slovakia.
- Iterative working method on continuous innovation strategy.
- Strong expertise in compression, especially adaptive and contextual.
- Certified as “Young Innovative Company” by the French Ministry of Research in 2013.
- Develops complex, high value-added products such as [GdPicture](#), [DocuVieware](#), and [PassportPDF](#).

Contents

- Optimization of existing PDF files: issues, constraints, and methods. 0
 - I - Context and issues 3
 - 1. Archiving 3
 - 2. Reduced storage costs 3
 - 3. Increase the speed of circulation in a collaborative work system..... 3
 - 4. Facilitate the transfer of attachments 4
 - 5. Ecological issue 4
 - II - Constraints 4
 - 1. The different standards 4
 - 2. The different legal or industrial requirements 4
 - 3. Can compression allow data loss? If so, to what extent? 5
 - III - Presentation of the format 5
 - 1. Overview 5
 - 2. Types of data contained in a PDF file: fonts, images, graphic objects, and more 6
 - 3. Different ways of embedding fonts 6
 - 4. The objects 6
 - 5. Support of different compression schemes..... 7
 - 6. Incremental backup 8
 - IV - Methods..... 9
 - 1. Lossless methods 9
 - 2. Methods with losses 12

I - Context and issues

1. Archiving

Archived documents generally do not have the same use as documents in circulation in a collaborative context. In this case, it will be interesting to optimize the documents once they have been generated and then distributed, but not before.

For example, during the release cycle of a document, different employees will use tools such as forms, annotations, attachments, and more. The set of data generated by these interactivity instruments will no longer be useful once the document is archived.

2. Reduced storage costs

A large number of PDF documents are stored. Some statistics:

In April 2016 (source: PDF Association, PDF Days Europe Berlin 2018)

- 2.2 billion PDFs on the web (Google search)
- 20 billion in Dropbox
- Airbus, Boeing and the US Department of Justice: more than 1 billion
- 2 billion PDFs open each year in outlook.com
- 73 million new PDFs saved each day in Google Drive and Mail
- 60% of attached files that are not images are PDFs in Outlook Exchange Enterprise

3. Increase the speed of circulation in a collaborative work system

Electronic invoices in PDF format represent by far the most considerable amount of documents in circulation today.

[Billentis](#) has published its annual e-billing report, this year called E-Invoicing/E-Billing. Digitalization & Automation. According to the study conducted by Bruno Koch and with the collaboration of EDICOM as a sponsor and expert in data integration systems, 550 billion electronic invoices are expected to be sent in 2019 around the world, with the volume of invoices in the narrow legal sense expected to quadruple by 2035.

4. Facilitate the transfer of attachments

- Not to exceed the size limit of the attached files on some platforms (Helpdesks in particular).
- Speed up the opening of PDF documents from mobiles.

5. Ecological issue

The cost of storage drops. But given the increasing amount of data to be backed up, the growing legal obligations of shelf life, as well as the aspects related to the redundancy of the latter (especially geo-replication), the resources needed to maintain their integrity, are exponential.

Technologies have their limitations. For instance, the evolution of the frequencies of microprocessors has been stopped abruptly for a few years now, forcing the manufacturers to find innovations in the architectures and the instruction sets of the latter to maintain a cycle of evolution.

Some research already reports a critical mass soon to be reached.

The storage space is like the air we breathe, and it will be given a real value when we will be threatened to miss it.

II – Constraints

1. The different standards

There are many standards in PDF today. Not all offer the same possibilities; some prohibit the use of existing features in earlier versions. It will be a question of adapting its strategy according to the targeted norm.

2. The different legal or industrial requirements

Many German software companies refer to PDF/A 1-b as the current industry standard for long-term archiving. This standard dates back to 2005 and is based on version 1.4 of the format and forbids, among other things, 8-bit transparency masks (Smaks), most non-embedded fonts, audio and video contents, JavaScript actions, and more.

Since March 2016, the German Federal Office of Information Security (BSI), which is part of the portfolio of the Federal Ministry of the Interior, has banned the use of correspondence matching and substitution

(Pattern matching & substitution - PMS) in data compression¹. The JBIG2 compression algorithm, when used to compress lossy data, uses this kind of technique. Although the ban does not target the use of JBIG2 compression, many publishers and service providers have decided to ban the JBIG2 encoder. Others will choose to use JBIG2 compression only as part of lossless compression. We will see in part 4 the motivations of this prohibition.

3. Can compression allow data loss? If so, to what extent?

We will introduce several methods of compression, some without loss of data, others with degradation. For this second category, it will be necessary to decide in advance whether the loss of data is tolerable, if so, to what extent.

Lossy compression will mostly be related to image reprocessing. It will be necessary for each image to decide if one can:

- Change its color depth. IE: 24-bit to 8-bit or 1-bit per pixel
- Perform a downscaling
- Alter its pixels (noise suppression, trimming, MRC processing ...)
- Re-encode with lossy compression algorithm (JPEG-2000 - JBIG2)

III - Presentation of the format

1. Overview

PDF (Portable Document Format) is a document format, developed by Adobe in 1992 and standardized by ISO in 2008.

It can be defined as a set of standards allowing the description of a document (of its pages as well as various other elements such as bookmarks, annotations, hypertext links, etc.).

The PDF format inherits from the PostScript language and uses the features of the latter to describe the text and graphic elements to present the information the same way on various applications and platforms.

¹ Ref: <https://www.pdfa.org/new/german-tr-resiscan-bans-lossy-jbig2-for-legally-significant-content/>

2. Types of data contained in a PDF file: fonts, images, graphic objects, and more

In order to fulfill this objective, each PDF file produced may include a set of digital data such as embedded files (images, fonts, color profiles) as well as description information based on precise data structures (bookmarks, annotations, destinations, color palette for indexed images, Cmap (character map), etc.).

3. Different ways of embedding fonts

Depending on the version of the PDF document produced, it will be possible to incorporate within it the font files, totally or partially.

Total: the file describing the font is embedded in its entirety.

Partial: A 'new' font will be generated to contain only the description of the characters used in the file.

If the file is to be edited during its life cycle, it may be better to incorporate the font in its entirety. Note that the font Arial Unicode weights 22 megabytes under Windows 10.

4. The objects

A PDF document is composed of various data structures, described as different objects. There are eight types of objects defined:

- Booleans (Boolean Objects)
- Numbers (Numeric Objects)
- Strings (String Objects)
- Names (Names Objects)
- Tables (Array Objects)
- Dictionaries (Dictionary Objects)
- Streams
- The null objects (Null Object)
- The reference objects (Indirect Objects)

In our interest in optimization, one is particularly important: the stream object. This object aims to store a data stream, usually binary (the content of an image, font file, color profile, embedded file, etc.), and offers various filters to compress the data it encapsulates. This is the only object whose contents can be compressed.

Version 1.5 of the PDF format introduces a new type of "Stream" object: ObjStm (Object Stream). The purpose of this type of object is to allow the compression of PDF objects that are not of the "Stream" type, which considerably reduces the size of PDF files.

We will see in the fourth section a practical case highlighting its interest.

5. Support of different compression schemes

The various PDF standards can offer up to 7 compression filters. Each will have its particularities. They will be used to encode the content of objects of types Stream and ObjStm. Depending on the standard chosen, the nature of the object to be compressed and the tolerance to loss, we can select a suitable filter. We will briefly introduce these filters in the next chapter.

The PDF specification allows seven compression schemes for images, which are:

- LZW (LZWDecode - PDF 1.0). Adaptive compression method, lossless created in 1984. Originally patented by Unisys and mainly used in GIF and TIFF digital image format. It is an improvement of the LZ78 algorithm that was created in 1978.
- RLE (RunLengthDecode - PDF 1.0). Lossless compression method, especially used for Group 3 and 4 faxes (black and white), BMP and PCX.
- CCITT (CCITTFaxDecode - PDF 1.0). Lossless compression method, for bitonal images only. It defines three different algorithms: Group 3 One-Dimensional (G31D), Group 3 Two-Dimensional (G32D), Group 4 Two-Dimensional (G42D). Standardized in 1988.
- JPEG (DCTDecode - PDF 1.0). Compression method typically used with loss, for 8-bit grayscale or 24-bit color images. Standardized in 1992.
- zlib / deflate. (FlateDecode - PDF 1.2 - 1996). A lossless compression method that couples the LZ77 algorithm and Huffman coding. Standardized in May 1996.
- JBIG2. (JBIG2Decode - PDF 1.4 - 2001). A compression method which can be lossless or lossless for bitonal images only. The JBIG2 algorithm produces files that are three to five times smaller than those compressed with the CCITT4 method. Standardized in 2001.
- JPEG 2000. (JPXDecode - PDF 1.5 - 2003). Compression method commonly used with loss, for 8-bit grayscale or color images using wavelet transforms (mathematical signal analysis method). This method generally produces higher compression performance than JPEG with sharper outlines and better-preserved contrasts. Standardized in December 2000.

Schéma	Norme PDF	Année	Avec pertes	Sans perte
LZW	1.0	1984		✓
RLE	1.0	1967		✓
CCIT	1.0	1988		✓
JPEG	1.0	1992	✓	✓
Zlib/Deflate	1.2	1996		✓
JBIG 2	1.4	2001	✓	✓
JPEG 2000	1.5	2000	✓	✓

The different compression schemes supported in the PDF specification.

6. Incremental backup

Version 1.4 of the PDF specification introduces support for incremental updates. Incremental updates provide a method for updating a PDF file without completely rewriting it. Thus, the content of a PDF file can be updated gradually without the need to regenerate existing data. The changes are added to the end of the file, leaving the original content unaltered. This type of backup is widely used in collaborative management systems to allow annotation and modify data very quickly while preserving earlier versions of the document within the same file.

IV – Methods

1. Lossless methods

1.1. Deleting content deemed unnecessary (attachments, forms, annotations bookmarks, etc.)

It is the most obvious approach. Once archived, it may not be useful at all to keep some data from a file such as:

- The attachments.
- The metadata, which can sometimes be very bulky because they can also include any type of data (photos, files, and more). Be careful; it is possible that metadata contain information useful for indexing the document if necessary. Precision: we can have metadata at different levels: document, pages, objects.
- The contents of interactivities, such as forms and JavaScript actions.
- The contents intended for the printing chain, such as the color profiles used by the printers.
- Bookmarks and hyperlinks.

1.2. Deleting unused objects and unused content


As described in the previous chapter, you can make changes to a PDF file incrementally. The counterpart of this approach is that the weight of the document is continuously increasing, even in the context of data deletion. For example, if a user wants to delete the last page of an existing file, the contents of that file will remain in a new product file. It may, therefore, be very efficient to regenerate a new file that will delete all the objects that are not used in the latest version of the generated document as long as:

- Preservation of previous versions is not required.
- The file does not contain an electronic certificate intended to validate the integrity of the document.

It is also not uncommon to end up with files containing unexploited objects even if no incremental backup has been performed. Indeed, the complexity of the format can result in the tool used to produce the file is found to integrate unused data.


1.3. Compression of objects of type Stream

We saw in the "format presentation" section that each Stream object could represent the data in a compressed way. However, it is not uncommon for applications that produce PDF files not to use this possibility. It will be appropriate to regenerate the file by compressing the data of these objects.

Fichier	Stream non compressé	Stream compressé
	<p>726 KB</p>	<p>63 KB</p>

1.4. Compression of other types of objects in sequence into an "Object Streams" structure introduced in version 1.5 of the format

Many PDF file generators do not use the "Stream / ObjStm" object types and therefore do not compress all the embedded data in the generated files. As a reminder, it is possible from version 1.5 of the specification to describe and group objects of Boolean types, numbers, strings, names, dictionaries and references in objects of type "Stream / ObjStm." It is easy to determine if the produced file uses this feature because with a simple text editor you can easily check that the version of the file is greater than or equal to 1.5 and observe if data of types previously listed are "readable." In this case, it will be convenient to regenerate the file by grouping and compressing the types of objects that are not of type "Stream" in "Stream / ObjStm" objects.

Fichier	Stream non compressé	Stream compressé
	<p>54.5 MB</p>	<p>15.69 MB</p>

Issues:

- Limit the number of serialized objects so as not to slow down the sequential access to the data. Adobe recommends a limit of 200 objects for non-linearized files and 100 objects for linearized files.
- Group the objects that will achieve an optimal compression ratio. An approach by nature can be effective at the first level. Some operational research should also provide adaptive serialization to increase performance and compression.


1.5. Deduplication of fonts

Often a document contains several times the same version of a font, especially when it is the result of the merger of several other documents.

The font deduplication optimization process can identify occurrences of the same font within a single file, and in some cases can produce significant results.

1.6. Converting embedded fonts into partial character sets

If the generated document is not intended to be modified, it may be interesting to turn the embedded fonts into partial character sets once the deduplication process is complete. It will be a question of recreating a font file which will integrate only the data related to the characters used².

Fichier	Poids original	Poids après optimisation
	<p style="text-align: center;">527 KB</p>	<p style="text-align: center;">34 KB</p>

² For further reading: <https://www.gdpicture.com/blog/pack-optimize-fonts-pdf/>
<https://www.gdpicture.com/blog/fonts-optimization-pdf/>

2. Methods with losses

2.1. Change the compression scheme for images (usually with loss)

It may be interesting, if necessary, to recompress images contained in an existing PDF file with a compression scheme to obtain a (much) better compression ratio.

As discussed in the format presentation section, the PDF specification allows for seven compression schemes, all of which can be used to compress images.

Among these different compression schemes, two offer particularly interesting optimization opportunities: JBIG2 methods for bitonal images (usually black and white) and JPEG2000 for 24-bit color and 8-bit grayscale images.

:: JPEG 2000

Pros:

- Compression often much better than all other schemes (for color images).
- The scheme allows choosing a compression ratio adjustment, which will vary the quality of the image produced.
- Offers very interesting results in terms of quality on colored text as part of an MRC compression.

Cons:

- Not available for PDF standards below 1.5, including the PDF / A-1a and PDF / A-1b standards imposed as standard for exchange and retention by some jurisdictions.
- Decompression can be very slow, especially on very high definition images which can alter the user experience with some viewers.
- Provides little compression/quality benefit on small images, especially photos.


Suitable for:

- Photographs.
- Structured documents scanned in color (invoices, forms, etc.).

Not suitable for:

- Compression of transparency masks. (The PDF specification makes it necessary to isolate the alpha channel from images with transparency in an image acting as a mask).

- Images with flat colors (logos, lines, geometric shapes with filling).
- Any other image whose loss is not desired.

Fichier	Résolution	Poids JPEG+ZIP	Poids JPEG-2000
	300 PPP	503 KB	426 KB

:: JBIG2

Pros:

- Compression can be lossless.
- Compression rate often much higher than CCITT4. Even in lossless mode.
- Can be used since version 1.4 of the PDF format.
- Supported in all PDF / A versions.

Cons:

- Lossy encoding³ can produce very unexpected results. CF problem photocopiers Xerox of 2013.


Suitable for:

- Since the lossless encoding is used, this scheme is suitable for all types of bitonal images.

Not suitable for:

- Lossy compression of sensitive data, the slightest substitution of symbols could cause problems.

³ The least desirable effect is character substitution. To date, no open-source library seems to be producing a correct result, even with a high desired level of quality. It will, therefore, be wise to ensure that the editor that develops the encoding can limit as much as possible this substitution based on heuristics or optical character recognition. If the targeted standard does not allow JPEG-2000 encoding, it will be interesting to serialize JPEG and Deflate compression. Indeed, the specification enables certain use cases of several filters. This practice provides compression gains of up to 15%.

Fichier	Résolution	Poids CCITT	Poids JBIG2
	300 PPP	227 KB	133 KB

2.2. Resizing images

Resizing images can provide significant gains in occupancy, but it is also a pretty destructive approach. There are many sophisticated resizing algorithms, but it will be rare to obtain a result that does not significantly degrade the perception of the image.

An interesting alternative may be to resize only the chrominance channel. It is the sub-sampling method of chrominance. Indeed, certain color spaces make a separation between brightness and color information. Since human vision is less sensitive to color than to brightness, it will be appropriate to reduce (or undersample) chrominance information without degrading the perceived quality of the image.

The jpeg and jpeg2000 compression schemes allow you to use this chrominance downsampling method.

- This method makes it possible to obtain an average space-saving of the order of 20%.
- 60% of images encoded in jpeg do not use this method.

2.3. Color detection

Color detection allows for the identification of images perceived as being in black and white or grayscale and has been incorporated with a color space using 24 bits (or more) per pixel. Although theoretically destructive of information, this approach generally produces excellent results in terms of quality and compression rate improvement.

2.4. Mixed raster content (MRC)

MRC compression, also known as Hyper Compression, is an image compression method that is particularly suitable for those containing text and continuous-tone graphics.

This method can reduce the size of images up to 8 to 10 times compared to JPEG in some cases it also improves the quality of the rendering of documents.

It will combine many techniques, among which segmentation will play a key role. Indeed, it will be mainly to separate certain regions of an image into several other images.


The classic approach to this type of compression is to create three separate images (or layers) from an image. Each of these layers will then be altered and optimally compressed. The specification of the PDF format will allow us to reconstruct the original document using specific rendering instructions.

We will designate these three layers like this:

- The binary layer. We will try to place the text and graphic elements in your uniforms (ex: lines, flat areas). This layer will ideally be saved in high resolution and compressed with the JBIG2 algorithm.
- The background layer. It will remain from the original image once the content for the previous layer is removed. The resolution of this layer will continue to be defined, by the user or using decision algorithms, and will be ideally recorded with the JPEG 2000 compression algorithm.
- The foreground layer. It is a layer that will contain the colors of the binary layer. It is, in a way, the equivalent of the chromatic channel of this one. As for the JPEG and JPEG-2000 compression, we can significantly reduce the resolution of this image that we will compress ideally with JPEG-2000 compression scheme.

Steps of an MRC engine:

- Pretreatment of the image. The goal is to improve the accuracy of the next step.
- Segmentation: detection of paragraphs, lines, words, characters, graphic elements.
- Analysis of each of the segmented elements to place them on the ideal layer.
- Post-treatment. The goal is to improve the compression of each layer as well as the quality of the final rendering (noise suppression, contrast adjustment).
- Compression of each layer.

Fichier d'origine (image JPEG)	Poids d'origine	Poids PDF/JPEG+ZLIB	Poids PDF/JPEG-2000	Poids PDF/MRC haute qualité
	686 KB	503 KB	426 KB	69 KB

2.5. Other techniques

- Reduction of the graphic objects description:

Especially for the figures generated from the CAD system. We can try to delete points on paths, reduce the number of useless decimals in floating-point numbers, convert smoothed lines into curves, etc.

- Flattening of transparency:

Flattening incorporates transparency into the corresponding artwork by splitting it into vector areas and pixelated areas.