PDF/A: digital documents to withstand the sands of time

Everything on the timeless standard for archived documents

PDF/A: digital documents to withstand the sands of time

Table of contents

PDF/A: The timeless standard for archived documents

While the carrier has transformed from physical to digital, archiving documents remains an essential and inevitable part of today's document management. For various industries and authorities it is even a legislated requirement. Often documents need to be stored for decades. And if you are going to go through all that trouble, you might as well go for a solution that does so consistently.

When it comes to the format and standard to use for digital archiving, there is a widespread acceptance of one specific standard: PDF/A. And for good reason. In this ebook we will look into those reasons, the standard's details, popular use cases, areas of application over various industries and an archiving case study of an implementation of iText by medical imaging specialist Zeiss.

We'll even take it one step further and provide you a tutorial on creating PDF/A conformant documents with the iText 7 Suite - a leading open-source PDF library/SDK. iText was the first to successfully bring PDF to the backend and has been doing so for over 20 years. As an active member of both the ISO-committee and the PDF Association, iText remains an innovator in PDF technology to this day and its software is deeply intertwined with the evolution of the PDF/A standard.

1.

Why PDF/A?

Why PDF/A is the widespread standard for archiving

PDF was originally developed by Adobe in 1993 to share, view and print documents in a visually predictable way, independent of the platform or software the user choses to view or manipulate the document with. While early adoption was slow, today PDF has become the most-used document format for online documents.

Then why not just use the general PDF standard for archiving?

PDF/A is different from PDF in the sense that in addition to platform independence, it enforces some extra restrictions to guarantee the document also remains consistent over longer periods of time. Since its introduction as an ISO standard in 2005 it has become the widely accepted standard for archiving.

To understand how PDF/A is different from PDF, we need to zoom in on those restrictions and requirements.

- A key aspect of PDF/A is self-containment, meaning **all content and info should be embedded** in the file. This includes the displayed content, fonts and color information (ICC color profiles).
- **Video and audio content is not allowed**, since these rely on external software to be rendered.
- **Encryption is not allowed**
- **Most JavaScript and executable file launches are forbidden** since they could alter the content of the PDF.
- **Standardized metadata** using the XMP format is required. This metadata can hold (but is not limited to) copyright information and the indication that the PDF is a PDF/A.

1.1 BENEFITS

Portable: PDF is platform-independent. This not only means that PDF documents can be opened on a wide range of devices, but that they will also be displayed in a consistent manner.

Non-proprietary: unlike other popular document formats, PDF is non-proprietary since Adobe released PDF as an open standard in 2008. As well as ensuring a wide-spread user base this also guarantees the availability of free tools.

ISO standard: PDF/A is defined in the ISO 19005 series of international standards. ISO is an independent international non-governmental organization ensuring user confidence in the quality, consistency and safety of the standard.

Searchable: XMP-metadata provides a consistent way of adding additional information to a document, such as the author, a description of the content, or its source and copyright. And using OCR (Optical Character Recognition) at the creation stage we can ensure that also all content in the PDF/A file is text searchable.

Is it always a good idea to use PDF/A?

Given that PDF/A is a subset of PDF with extra restrictions on what it can contain, it is important that a user is aware of what is and isn't allowed when creating PDF/A documents. This is especially true when converting from an existing digital document. For example, non-embedded fonts might not be available to embed on the machine or forbidden content such as video or JavaScript will be removed in the process. This risk is especially inherent when automating the migration, as is often the case with archiving.

In addition to fully understanding the PDF/A standard (which you will after reading the next chapter), PDF/A validation can also offer some reassurance. Although it is important to note that PDF validation doesn't guarantee valid PDF, since it only focusses on the requirements in the PDF/A ISO standard (ISO 19005).

Later in this ebook we will zoom in on how a hybrid form of archiving within the boundaries of the PDF/A-3 subset can mitigate some of this risk.

A PDF can contain much more than just static text and images. When creating or converting to PDF/A we have to be aware of what its restrictions mean for our content.

1.2. WHAT ABOUT TIFF?

The TIFF format was created in the mid-1980s as a standard file format for storage of scanned images. While TIFF is often seen as just another image format, it is much more than that: it is a fully-fledged file format which can act as a container for different image files. It was the go-to for digital archiving of documents before PDF was around, but it was never intended to serve this purpose. Today TIFF is ousted by PDF/A in all but image-only use cases. The main disadvantages of TIFF:

- Since TIFF files contain no text, the contents are not text-searchable
- Multi-page documents can be extremely large, and if documents contain color images it becomes all but impossible to compress them efficiently within the TIFF format
- While historically widespread, TIFF's longevity is not guaranteed, and TIFF as a whole is not an ISO-standard.

2.

Understanding PDF/A

What makes a document PDF/A

Now that we understand how PDF/A is a tailored subset of the general PDF standard, making it perfect for archiving, it is time to zoom in on the different parts and conformance levels within the PDF/A standard.

2.1 PARTS AND CONFORMANCE LEVELS

The PDF/A standard consist of four different parts with their associated conformance levels. While parts one to four were chronologically released, later parts shouldn't be seen as purely better versions but rather as an extension of the archiving options prior parts offered. In general terms, later parts are less restrictive in the sense that they allow more, and sometimes new PDF capabilities introduced in the general PDF version they are based on.

PDF/A				
	PDF/A-1	**PDF/A-2**	**PDF/A-3**	**PDF/A-4**
ISO-standard	ISO 19005-1:2005	ISO 19005-2:2011	ISO 19005-3:2012	ISO 19005-4:2020
Based on	PDF 1.4	PDF 1.7	PDF 1.7	PDF 2.0
Published in	2005	2011	2012	2020
Conformance levels	a,b	a,b,u	a,b,u	e,f

PDF/A-1 is a subset of PDF 1.4, whereas PDF/A-2 and PDF/A-3 are a subset of PDF 1.7's ISO 32000-1 standard. In 2020 PDF/A-4 was released based on PDF 2.0.

2.1.1 PDF/A-1

PDF/A-1 is defined in ISO 19005-1:2005.

The first part of the PDF/A standard was published in 2005 and is based on PDF 1.4. PDF/A-1 can be seen as the strictest of the four parts. It forbids:

- Transparent elements
- Layers
- JPEG2000 and LZW compression

2.1.2 PDF/A-2

PDF/A-2 is defined in ISO 19005-2:2011.

PDFA-2 was published in 2011 and is based on PDF 1.7, making use of many of its newly introduced features over PDF 1.4. Moreover, the restricted features (transparency, layers and JPEG2000 and LZW compression) in PDF/A-1 are introduced and allowed here. With PDF/A-2, OpenType Fonts can now also be embedded and support for PAdES-compliant signatures was added.

Especially interesting is the possibility to embed other PDF/A files (PDF/A-1 or PDF/A-2 to be precise) within the PDF/A-2 document, making it an effective container for multiple documents.

2.1.3 PDF/A-3

PDF/A-3 is defined in ISO 19005-3:2012.

PDF/A-3 was introduced in 2012 and like PDF/A-2 is based on PDF 1.7. But while not introducing new features to the format itself, it introduces one interesting and much discussed functionality that was strictly forbidden in prior versions: one can now embed files of **any** type in a PDF/A-3 document.

This was much discussed in the sense that it raised some concerns within the archival community as this could be a way to bypass restrictions on document formats when the PDF/A-3 format is used as a (nearly) empty shell for non-conformant file types.

However, when used responsibly, PDF/A-3 permits an interesting new form of **hybrid archiving.** This entails embedding the original document the PDF/A-3 was

created from. The main advantage to this approach is that it mitigates loss of data in the migration process by having the original data available.

A PDF can also act as a container for multiple files and file formats and this functionality is known as “portable collections” in the PDF specification (or PDF portfolios in Adobe Acrobat), and has several other use cases other than hybrid archiving. We expand on such use cases in our article about PDF portfolios and how to use them. Keep in mind that the embedded files do not necessarily adhere to the strict PDF/A standards and therefore their longevity and consistency cannot be guaranteed.

Another notable use is the attachment of a **machine-readable (XML) copy** to the PDF/A-3 document. This allows external applications to process the content in a much more performant way. This is famously applied in the ZUGFeRD specification, a format for electronic invoices that combines a PDF/A-3 for visual representation with an XML-file for machine interpretation.

2.1.4 PDF/A-4

PDF/A-4 is defined in ISO 19005-4:2020.

Published in 2020, part 4 of the standard is based on PDF 2.0 and allows some of its new features such as page level output intents (output intent tells the processor how to interpret the colors used in the document).

Another big difference is that it drops the A, B and U conformance levels (more on conformance levels in the following subsection) and introduces two profiles that extend the general PDF/A-4 spec:

- PDF/A-4f for embedding any other file, making it a successor to PDF/A-3.
- PDF/A-4e for support of Rich Media and 3D annotations

PDF/A-4 is also the only part of the specification to allow JavaScript. To prevent harmful use and unintended alteration of content, it is however only to be stored in an embedded file stream and shouldn’t be executed by a viewer without explicit action by a user. The main goal here is to preserve the process by which a file reached its current state (if it’s an interactive form, for example), but in no way is the preservation of working JavaScript guaranteed.

2.1.5 Conformance levels

Level b ("basic"): ensures that the visual appearance of a document will be preserved for the long term.

Level a ("accessible"): ensures that the visual appearance of a document will be preserved for the long term, but also introduces structural and semantic properties. The PDF needs to be a Tagged PDF.

Level u ("Unicode"): ensures that the visual appearance of a document will be preserved for the long term, and that all text is stored in Unicode. This facilitates the searchability of text. As we will see in chapter 5.3 this is the preferred output format for OCR-generated PDFs.

The **f and e conformance levels** are much more functional profiles that extend the specification than conformance levels and are discussed in subchapter "2.1.4 PDF/A-4".

2.1.6 PDF/A and accessibility

How does PDF/A with a-level conformance (as in PDF/A-1a, PDF/A-2a and PDF/A-3a) compare to PDF/UA, the standard for universally accessible PDF?

Accessibility in PDF is of prime importance to users with a disability using assistive technology such as a screen reader. First and foremost a logical document structure is needed for assistive technology to interpret a document properly. In addition to disabled users, this structure is beneficial and needed for efficient machine reading such as that of search engines. This structure is implemented by tagging the different sections of PDF, i.e., Tagged PDF.

The a-level conformance in PDF/A is lacking in this broader sense of universal accessibility. Technically it is possible to comply with PDF/A-1a with as little as a single tag on each page, making it quite meaningless in terms of true accessibility to both human and machine.

In contrast, the PDF/UA standard requires full and meaningful tagged PDF. It also requires exclusion of problematic content where meaning is conveyed by the use of graphic elements or the use of color or contrast.

The reason for this discrepancy is that at the time of initial development of the PDF/A standard, a full description of what accessibility meant in PDF such as in the PDF/UA specifications was not around, and the desired accessibility was limited to guaranteeing that a document remained both visually and logically reproducible over time.

It is however recommended when creating a PDF/A a-level document to aim for it to also be PDF/UA compliant. This is for example recommended by the Library of Congress. Keep in mind that where encryption in PDF/UA documents is allowed, it is forbidden in the PDF/A standard.

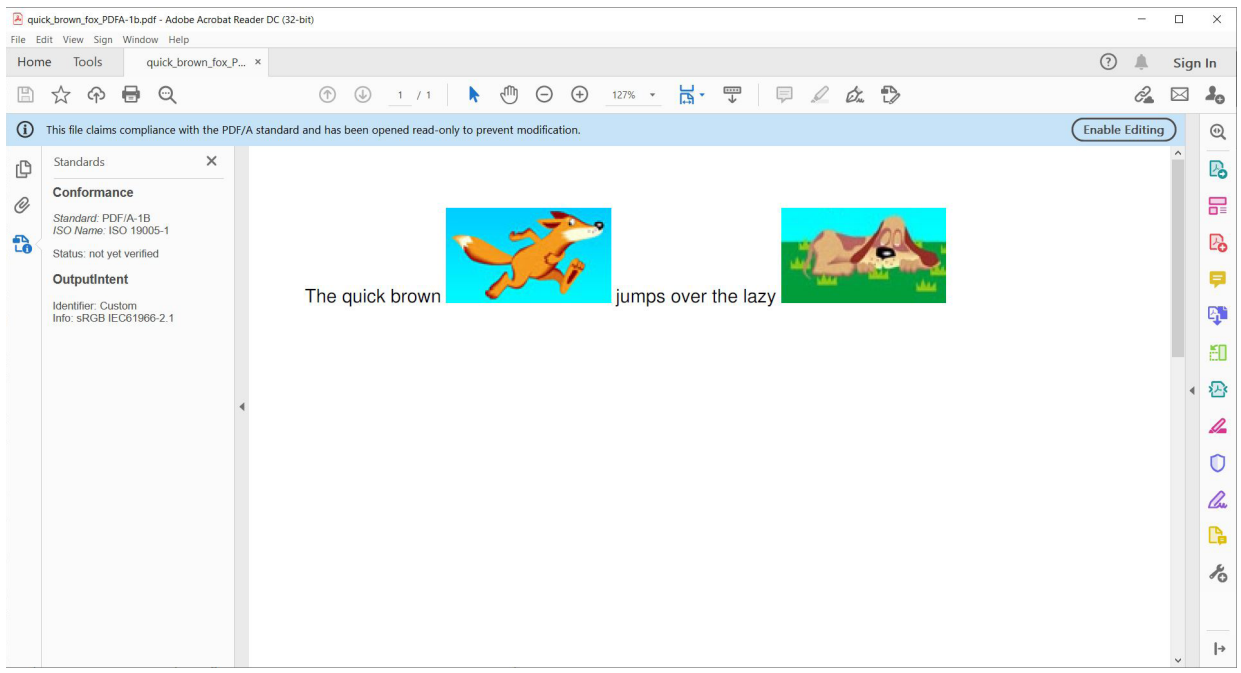
2.2 PDF/A VALIDATION

So now that we know what makes a PDF document PDF/A compliant, how do we check if this is indeed the case for the documents we create or receive?

This is not always easy to do at first sight. While popular applications such as Adobe Acrobat Reader feature a blue bar on top of documents that claim to be PDF/A, this no guarantee for compliance.

In order to check for full PDF/A compliance, we can use a validation tool. By far the most popular and reliable solution is veraPDF. This open-source solution was originally funded by the EU and is supported by the PDF community represented by the Open Preservation Foundation, the PDF Association and the Digital Preservation Coalition. Today veraPDF is maintained by the Open Preservation Foundation and developer Dual Lab.

iText Software's PDF library/SDK iText 7 doesn't contain a PDF validation tool, but instead recommends and supports veraPDF. Moreover, iText does use veraPDF in its test suite to prevent introducing any PDF/A conformance issues.



A document that **claims** to be PDF/A conformant, is indeed nothing more than a claim. PDF/A validation is needed to ensure compliance with the standard.

3.

Why PDF/A is relevant today

Areas of application and case study

The whole onset of PDF/A ensures in essence that it will remain forever relevant: it is a document format that preserves the visual appearance of the content over time. The biggest proof of that relevance lies in its widespread use.

Many industry-specific compliance regulations also offer a challenge that can often be resolved with the use of PDF/A.

In this Chapter we give an overview of the different areas of application for archiving within the PDF/A format and we zoom in on an archiving case study in healthcare featuring Zeiss and iText.

"50% of the Fortune 500 companies and 70 % of Fortune 50 companies use iText and thus PDF technology in their document workflow. When it comes to archiving, PDF/A is the obvious choice for them"

Raf Hens, CTO at iText Software

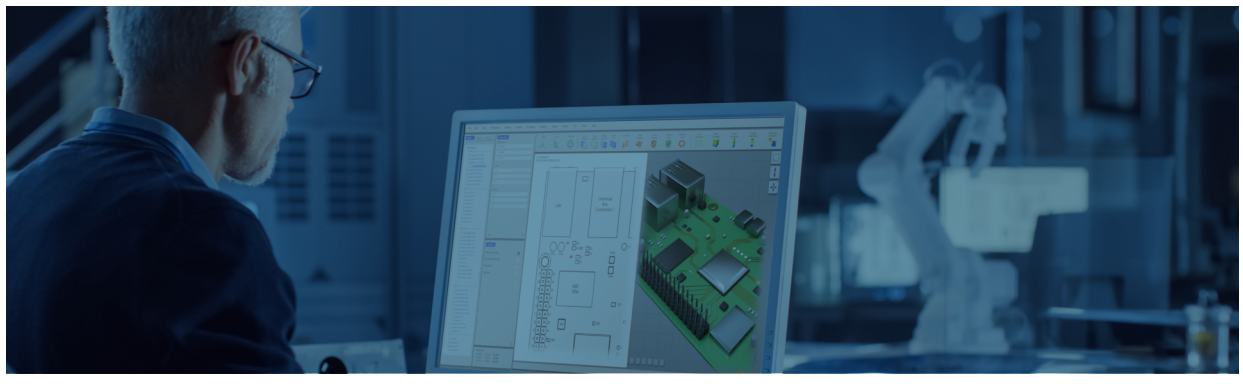
3.1 AREAS OF APPLICATION

3.1.1 Governments and public institution

Many public authorities recommend PDF/A and some even make it a hardline requirement. For example, the Dutch, Swiss and the Danish governments all enforce the use of PDF/A for non-editable documents.

Among archive institutions PDF/A is also one of the preferred formats for The Smithsonian, New York State Archives, the National Archives of the Netherlands, The National Archives of the UK, Standford University and so on.

3.1.2 Manufacturing and construction



Industry documentation often needs to be kept available for future reference and liability issues. A famous example is that of aeroplane manufacturer Airbus. Aeroplane blueprints must be preserved for at least 99 years. Even before PDF/A was created, the Airbus team developed a “minimal PDF” to avoid the pitfalls of general PDF.

PDF/A-3 has the added advantage that any sort of file such as 3D models can be embedded into the PDF container.

3.1.3 Financial sector

The financial and insurance sector sometimes requires documents to be retained for 50 or more years. Additionally, financial records and sensitive client information needs to be kept in a contained and secure format.

Another great use case is the one of e-invoicing. As we discussed earlier in our PDF/A-3 section, the ZUGFerd format uses the PDF/A-3 container capabilities to embed invoice data in a machine-readable XML format with the original invoice. This allows for appropriate software to process the invoice automatically.

3.1.4 Healthcare

As a general rule medical documents need to be preserved for up to 30 years. Think patient records, medical statements, reports and imagery like X-rays. Digital signing and time stamping is often added to attach an audit trail to documents.

Medical documents archived as PDF/A can also be a useful resource for research on long-term effect of medication and treatments. PDF/A assures a reliable visual representation for decades and with tools such as iText pdf2Data, data can be extracted in an intelligent and consistent way.

A PDF/A case study in healthcare – ZEISS: using iText to create and archive ophthalmic reports

Background

As specialists in ophthalmology, microsurgery and other medical growth sectors, ZEISS manufactures innovative products such as their range of ZEISS ophthalmology devices. These consist of products and solutions to enable efficient diagnosis and treatment of cataracts, glaucoma, and other retinal disorders.

Goals

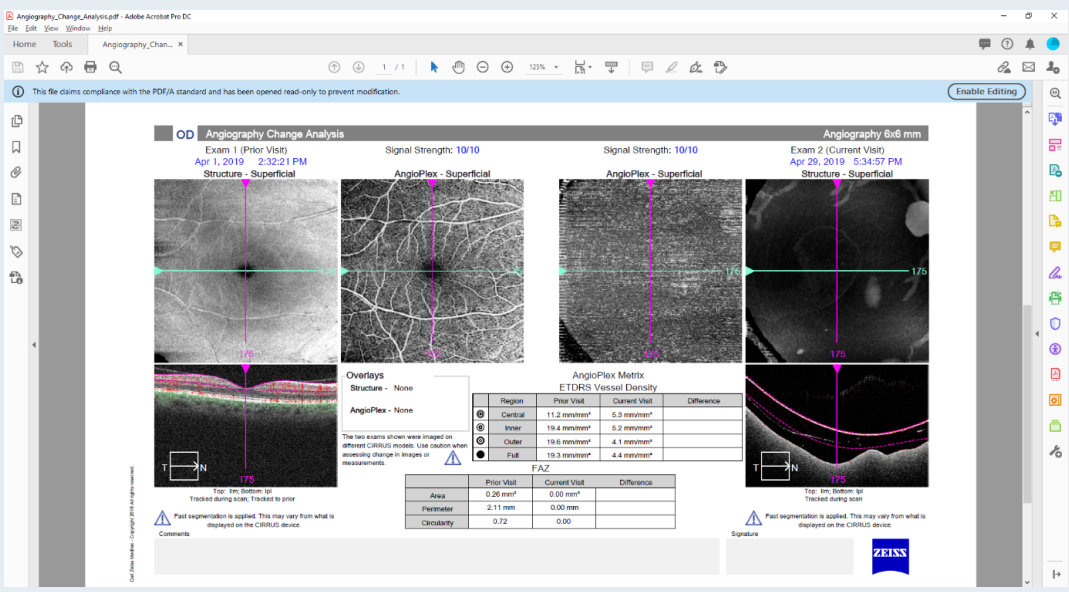
An important focus for ZEISS is the networking of systems and integrating data management to improve medical workflow efficiency. To enable this, ZEISS developed their FORUM family of software applications. FORUM is a scalable and flexible data management system that evaluates clinically relevant data from diagnostic devices and gives direct access to the full examination history of patients.

Challenges

- To allow FORUM to integrate data from both DICOM compliant and non-compliant devices into PDF reports
- To ensure the resulting PDF reports are PDF/A compliant

Offered solution

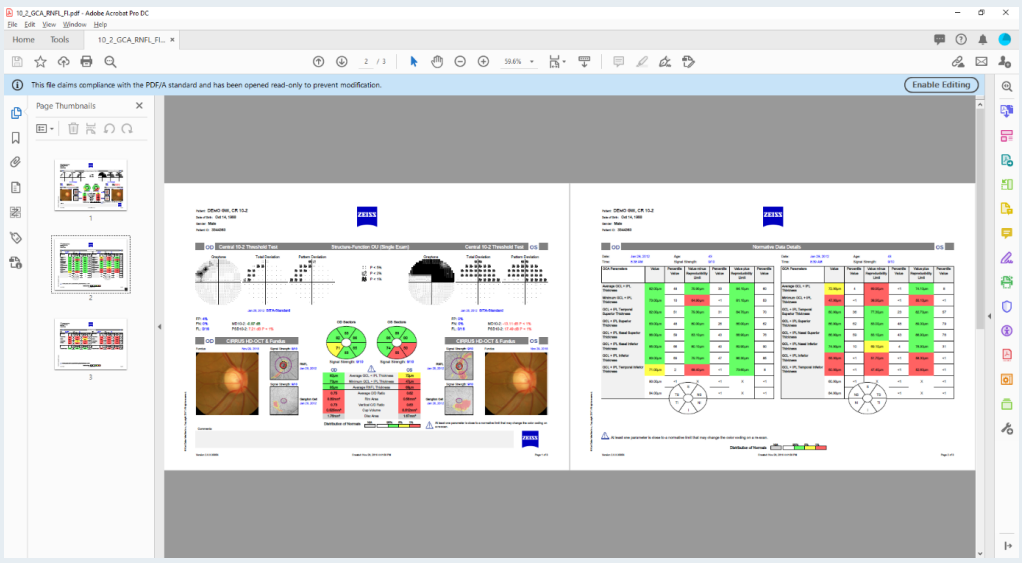
iText was chosen as the PDF generation engine for FORUM due to its strong PDF/A capabilities, a requirement for the DICOM (Digital Imaging and Communications in Medicine) standard which defines the formats for exchanging medical images.



An Angiography Change Analysis report produced by the FORUM software.

Patient diagnosis reports must comply with the DICOM standard to allow them to be shared and archived on a long-term basis. ZEISS has integrated iText in FORUM since its initial development in 2011, and it is a vital component of the application's functionality. FORUM integrates data from both DICOM compliant and non-compliant devices. When a patient undergoes an eye analysis such as a visual field examination or an angiography, iText is used to format and combine images and other data from the analysis into a PDF report.

iText can also combine data from multiple sources and create custom reports as required. The results can then be reviewed by medical professionals, and as reports are produced as PDF/A they can be archived as required by the DICOM regulations.



A combined PDF report produced by the FORUM software

"We're very happy with the PDF/A functionality provided by iText, and it forms an essential part of our DICOM-compliant reporting framework."

Robert Hien, Lead Software Developer: ZEISS

Result

PDF/A compliance is a particular strength of iText, and it fully supports all conformance levels of the PDF/A-1, PDF/A-2 and PDF/A-3 standards.

ZEISS recently upgraded to iText 7, taking advantage of its enhanced features such as the improved document model and layout engine. PDF generation in FORUM was developed using the .NET version of iText 7, though expanding support for generation in Java-based environments is being considered, thanks to the iText Java and .NET APIs being identical in the way functionality is implemented.

4.

Use cases

From paper and digital sources to PDF/A

4.1 SCANNED DOCUMENTS

Scanning and archiving paper documents was the initial use case PDF/A was brought to life for. It was and is primarily used in mailroom and records digitization. While we are shifting more and more to a complete digital flow, the process of digitization is an ongoing process that will persist for many years to come.

As we discussed in the first chapter scanning to PDF rather than to a simple image or TIFF file has numerous advantages:

- **High compression rate**: Thanks to a combination of efficient compression algorithms such as DEFLATE, JBIG2, JPG, JPG2000 and LZW (the last two are not allowed in PDF/A-1) used on the appropriate text and image layers.
- **Text-searchable**: With OCR (Optical Character Recognition) an image with text can be converted to searchable and editable text. Furthermore, we can add important metadata in the XMP format.
- **Digital signatures**: digital signatures can be added making PDF/A an excellent medium for archiving legally binding documents.

And of course, you unlock the full potential of further manipulation within the PDF format (within PDF/A's necessary limitations), for example redacting sensitive information or accurately rendering advanced typographic scripts.

4.2 DIGITAL-BORN DOCUMENTS

While with scanning paper document to PDF/A we can rightfully say there are only advantages, the use case for migrating digital-born documents of various resources to PDF/A is a bit more delicate. After all, if we are not careful, we can easily lose content such as embedded fonts. Of course, this is all dependent on the software you use to make the conversion and the rules you enforce in your document workflow, and not on the PDF/A format as such.

4.2.1 PDF to PDF/A

When migrating from the general PDF standard to PDF/A there are a number of things we need to take into consideration. While some of the specifications in PDF/A focused on omitting rich content are easy to implement, others can prove harder without human intervention:

- Metadata: converting Info Dictionary entries to the XMP format. A set of rules is required to do this properly.
- Embedded fonts: since all versions of PDF/A require embedded fonts, these need to be available at the time of conversion for embedding
- Conformance level a: migrating to conformance level a introduces the extra challenge of making a document tagged for improved accessibility. While software such as iText can tag a document automatically, this still requires the source document to have an acceptable structure to start from.

So be wary of applications guaranteeing a perfect automated conversion. As we have seen in the PDF/A validation section of this ebook, a simple blue bar on top of your document doesn't necessarily mean your document conforms to the PDF/A standard.

4.2.2 Office to PDF/A

Over 500 billion Office documents are created every year across multiple industries, yet they have significant drawbacks when it comes to archiving.

Many third-party tools such as the Microsoft Office suite have long implemented native PDF conversion. Unfortunately, most of the time there is no option to convert to the PDF/A specific parts such as PDF/A-2 or A-3, and many only comply to PDF/A-1b. Because of this shortcoming, and for the reason that migrating to PDF for archival purposes is mostly done in batch processes, we focus on automated solutions in this ebook.

One solution would be to run a batch process instantiating the application that can

render the source format and then convert it. Since this would happen sequentially, this is a highly inefficient process.

A better solution is to make use of a powerful PDF library to do the heavy lifting. This solution does not rely on external software to perform the conversion, everything is handled out of the box.

4.2.3 Email to PDF/A

Archiving email is a complex matter. In addition to plain text, content often start from a HTML email with a complex body and header, there is also no limit to the diversity of possible attachments.

One way to tackle these attachments is to apply rules in your workflow per file format that either migrate those files to a PDF/A conformant document or embed them in a PDF/A-3 file.

The header information should be converted into the PDF/A file's XMP metadata, helping with searchability.

The body is stored as the main content of the PDF/A file.

4.1.4 HTML to PDF/A

Converting to PDF/A from HTML is not as simple as simply generating it from your favorite browser. In order to easily create PDF/A we need to keep the structural information in the HTML intact.

And if we want to make the process performant when we apply it on a large scale, a **headless solution** is preferred. This means that the process of the conversion isn't graphically displayed, but rather runs in the background without the need for a substantial visual rendering overhead.

iText pdfHTML does just that for you, has PDF/A capabilities embedded and supports advanced HTML and CSS features.

5.

Making a PDF/A compliant document with iText 7

Using a leading PDF library to create PDF/A documents

We've discussed several use cases for when we would want to archive both scanned and digital-born documents, but how do we implement this?

The obvious drawbacks of using one or more instances of desktop apps make it clear that we need an automated code-based solution. Rather than getting into the elaborate PDF specifications and creating an implementation from scratch, we will look in to how we can use a powerful PDF library/SDK to create and manipulate PDF/A with just a few lines of code.

iText 7 offers you the most versatile and well-documented PDF library, written for Java and .NET (C#). With two decades of improvements and millions of users, our PDF library offers you an experience that is both robust and extensive. It can meet all the needs of your document workflow. From creating compliant PDF, to secure digital signing of documents. iText 7 does it all, and does it well.

Added to the ability to create PDFs that are fully compliant with the general PDF standard, iText 7 also has the ability to generate PDF/A (and PDF/UA) documents built into its core package. In this chapter we will focus on creating PDF/A documents with iText 7 Core and some of the manipulations we can perform on those documents with the iText 7 Suite, which comprises Core and its add-ons.

Note: Since iText 7 is constantly improving with four releases a year, all code samples will be referenced to our always up-to-date knowledge base.

5.1 CREATING A PDF/A-1 DOCUMENT

Resources:

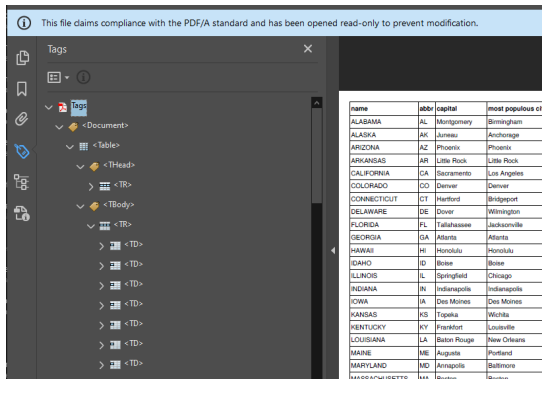
- **Guide:**
  - **Java:** https://kb.itextpdf.com/home/it7kb/ebooks/itext-7-jump-start-tutorial-for-java/chapter-7-creating-pdf-ua-and-pdf-a-documents
  - **.NET (C#):** https://kb.itextpdf.com/home/it7kb/ebooks/itext-7-jump-start-tutorial-for-net/chapter-7-creating-pdf-ua-and-pdf-a-documents-net
- **Code example PDF/A-1b:** https://kb.itextpdf.com/display/IT7KB/iText+7+jump-start+tutorial+%3A+Chapter+7#iText7jump-starttutorial:Chapter7-c07e02_quickbrownfox_pdfa_1b
- **Code example PDF/A-1a:** https://kb.itextpdf.com/home/it7kb/examples/itext-7-jump-start-tutorial-chapter-7

iText 7's open-source Core library gives us all the tools we need to make a PDF/A compliant document. Let us take a look at what that means for a PDF/A-1b document:

- We can **create a PDF-A document instance** where we can provide the **desired conformance level** as a parameter: in this case we will start off with PDF/A-1b. We can also set the output intent which defines the color usage.
- A PDF-A document is self-contained. For that reason, we **embed the font** we are using.
- The document needs to contain **metadata in the XMP format**. iText provides functionality to automatically translate the PDF's Info Dictionary to XMP.

When we want to make our document PDF/A-1a compliant we need to add some accessibility features:

- **Change the desired conformance level** to PDF/A-1a.
- Make the PDF a **Tagged PDF** with one line of code.
- Add alt text to the images by changing the appropriate accessibility property.



When we set tagging as required for a-level compliance, our document structure is clearly visible in the Tags panel.

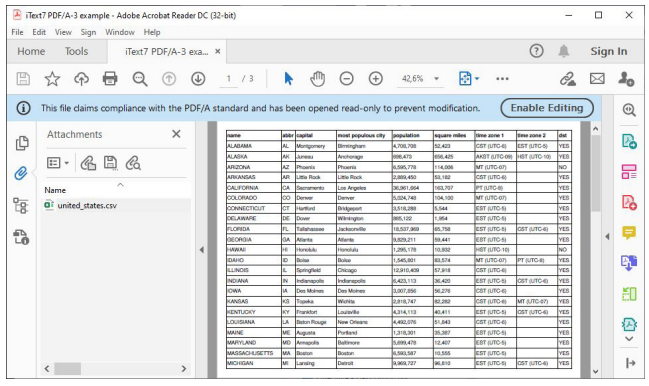
5.2 CREATING A PDF/A-2 AND PDF/A-3 DOCUMENT

Resources:

- **Guide:**
  - **Java:** https://kb.itextpdf.com/home/it7kb/ebooks/itext-7-jump-start-tutorial-for-java/chapter-7-creating-pdf-ua-and-pdf-a-documents
  - **.NET (C#):** https://kb.itextpdf.com/home/it7kb/ebooks/itext-7-jump-start-tutorial-for-net/chapter-7-creating-pdf-ua-and-pdf-a-documents-net
- **Code example:** https://kb.itextpdf.com/home/it7kb/examples/itext-7-jump-start-tutorial-chapter-7

For both PDF/A-2 and PDF/A-3 the biggest difference with a PDF/A-1 document is that we can include attachments. While for PDF/A-2 that can only be another PDF/A document, it can be any file type for PDF/A-3. For example, we could attach the original file that contains the data we use in our PDF.

In iText 7 Core we can add this file as an attachment with the appropriate parameters required for PDF/A-3. Note that the filename should be set to Unicode and that one of the parameters is a newly created PDF dictionary.



We can find the attached file in the attachments panel of Adobe Acrobat Reader.

5.3 MAKING A SCANNED PDF/A TEXT-SEARCHABLE WITH ITEXT PDFOCR

Resources:

- **Guide:** https://itextpdf.com/en/blog/technical-notes/how-use-itext-pdfocr-recognize-text-scanned-documents
- **Code example:** https://kb.itextpdf.com/home/it7kb/examples/pdfocr-how-to-ocr-an-image-to-pdf-a-3u

One of the major challenges in document management is dealing with inaccessible data, data which is locked away in non-editable documents. Scanning a document containing printed text does not make it editable or searchable however, you just have a scanned image of the content.

Optical Character Recognition (OCR) can help to unlock this data. One of the most common use cases for OCR is to produce documents which can be searched, processed and archived. While some word processing and PDF applications now offer OCR functionality to make PDFs editable, manually doing this for more than a few documents is impractical.

The open-source tool iText pdfOCR provides a way to automate the OCR process and integrate it into document workflows. It is powered by the popular Tesseract 4 open-source OCR engine and has the capability to output to the PDF/A-3u format best suited for OCR.

5.4 OTHER PDF CAPABILITIES WITH ITEXT 7

Digital signing

In order to have legal value, archived documents may require them to be signed. The core PDF standard supports digital signatures and they are allowed in all PDF/A parts. So, adding a digital signature does not invalidate a document's PDF/A conformance.

Parts PDF/A-2 and later require PAdES (PDF Advanced Electronic Signatures) compliance, a PDF-specific implementation of CAdES (CMS Advanced Electronic Signatures) which adds an advanced set of digital signature restrictions and extensions to PDF and ISO 32000-1.

iText has the latest digital signing functionality, including all the necessary PAdES parts, embedded in its open-source iText 7 Core library.

Extracting data

When you want to leverage your archived files to collect historical data, iText has you covered. iText 7 Core has data-extracting capabilities, but if you want to so in an intelligent way it is worth looking into iText pdf2Data.

iText pdf2Data is a solution to easily recognize and extract data from documents. It is available for Java and C# (.NET), and as a CLI version.

It offers a framework to intelligently recognize data inside PDF documents, based on selection rules that you define in a template. It has a visual template editor, so you don't need to be a developer to use this tool.

Support for languages with complex writing systems

pdfCalligraph is an iText 7 add-on for Java and C# (.NET) that allows you to unlock advanced typographic features in PDF. It also allows you to expand your document workflow with global languages and writing systems, that incorporate accurate rendering and are suitable for data processing.

This is a vital component when archiving documents that feature languages with these complex scripts and right-to-left writing systems.

Redacting

Keeping sensitive text and images inside a document classified can prove to be especially important in archiving, where files will remain accessible for decades.

pdfSweep is an iText 7 add-on for Java and C# (.NET) that removes (redacts) information from a PDF document in a reliable and secure way. If you are not familiar with redaction, it is those typical black bars covering part of a document's text you might have seen in a spy movie featuring classified documents. That's not all though as pdfSweep also allows redaction of images (or parts of an image).

Optimizing your archive for size and speed

Keeping files archived for decades also means paying for data-storage for decades. On your own servers or in the cloud, it is a costly affair. And if you want to access or manipulate these files, you don't want this to take ages. That is why it is important to have a performant and space-saving archive.

pdfOptimizer is an iText 7 add-on that lets you optimize your PDFs with a custom or pre-fit archiving profile. The great news is that with the intelligent options in pdfOptimizer you can do this while leaving the visual appearance untouched:

- Intelligent compression of images: different images require different compression and scaling techniques. Some of them can be applied without the user noticing. Note that PDF/A-1 does not allow for JPEG2000 and LZW compression.
- Removing duplicates: remove duplicate instances of embedded fonts and images.
- Font subsetting: removing unused characters of a font.
- Stream compression: binary streams can be compressed without quality loss in the generated files.
- Compress attachments

Font optimizations such as removing duplicates and font-subsetting can prove to be an important space saver for PDF/A documents since all fonts need to be embedded.

Flatten forms

All PDF/A documents forbid dynamic content (except for PDF/A-4, but even then it is restricted). So, what if we want to archive a PDF containing XFA forms?

pdfXFA is an iText 7 add-on for Java and C# (.NET) that allows you to flatten dynamic XFA forms to static PDF.

When something goes wrong

Organizations and governments often have to archive documents coming from a wide array of sources. This also means that the quality of the initial PDF cannot always be guaranteed, and your PDF/A conversion might end up looking different as you expected.

iText RUPS is a diagnostic tool for reading and updating PDF Syntax that allows you to have a crystal-clear view on the inner workings of your PDF.

ABOUT US

Get to know iText

Your boarding pass for your flight. Or an invoice, receipt or form in a PDF format.... Most likely they were generated by iText technology!

iText is a global leader in innovative award-winning PDF software. It is used by millions of users - both open source and commercial - around the world to create digital documents for a variety of purposes: invoices, credit card statements, mobile boarding passes, legal archiving and more.

iText works and works well. Our customers choose iText because of our world-class software quality, and our reliable, mature, and proven technology. We are recognized as a global thought leader and innovator in PDF solutions and functionalities. Our PDF solutions can be embedded into the document workflows of various industries and their applications to enable creation and manipulation of PDFs, and advanced features like secure content redaction, encryption, digital signatures, and ensuring documents are accessible and archivable.

Our diverse customer base includes many of the Fortune 500 companies, as well as small companies and government agencies. We strongly believe in the value of open-source software. Our core library, iText 7, is available under the AGPL license. We also offer commercial licensing for customers that do not wish to comply with AGPL and want to keep their source code private.

VISION

In a world in which speed and efficiency are paramount, we enable companies and people to build the most reliable solutions for document and data exchange, effortlessly.

MISSION

It’s our mission to be the most trusted and comprehensive technology provider which perfectly leverages the power of PDF, by offering open-source and enterprise solutions that streamline the generation and consumption of documents and data.

CONTACT

marketing@itextpdf.com
www.itextpdf.com

OUR OFFICES

EUROPE, MIDDLE EAST, AFRICA & CIS

AA Tower
Technologiepark-Zwijnaarde 122
9052 Zwijnaarde
Belgium

sales.isb@itextpdf.com
Tel +32 9 298 02 31
Fax +32 9 270 33 75

AMERICAS

530 Harrison Ave,
Second Floor
Boston, MA 02118
United States

sales.isc@itextpdf.com
Tel +1 617 982 2646
Fax +1 617 982 2647

ASIA & OCEANIA

Republic Plaza
9 Raffles Place, Level 6, Republic Plaza 1
SINGAPORE 048619
Singapore

sales.isa@itextpdf.com
Tel +65 6932 5062

itextpdf.com