

# Whitepaper: PDF/A with PDFlib Products

*The PDF/A formats specified in the ISO 19005 standard strive to provide a consistent and robust subset of PDF which can safely be archived over a long period of time, or used for reliable data exchange in enterprise and government environments. This whitepaper discusses PDFlib features for creating PDF/A output suitable for long-term document archival.*

**PDF/A-1a and PDF/A-1b.** PDF/A-1, formally the international standard ISO 19005-1, is targeted at reliable long-time preservation of digital documents. The standard is based on PDF 1.4 and imposes some restrictions regarding the use of color, fonts, annotations, and other elements. There are two flavors of PDF/A-1, both of which can be created and processed with PDFlib products:

- ▶ ISO 19005-1 Level B conformance (PDF/A-1b) ensures that the visual appearance of a document is preservable over the long term. Simply put, PDF/A-1b ensures that the document will look the same when it is viewed or printed some time in the future.
- ▶ ISO 19005-1 Level A conformance (PDF/A-1a) is based on level B, but adds crucial properties from »Tagged PDF«: it requires structure information and reliable text semantics in order to preserve the document's logical structure and natural reading order. Simply put, PDF/A-1a not only ensures that the document will look the same when it is used in the future, but also that its contents (semantics) can be reliably interpreted and will be accessible to physically impaired users.

When PDF/A without any conformance level is mentioned below, both conformance levels are meant. The PDF/A implementation in all PDFlib products is based on ISO 19005-1:2005 plus Technical Corrigendum 1 (2007).

**PDF/A requirements and restrictions.** PDF/A requires certain PDF features and prohibits certain others. For example, in order to guarantee exact text reproduction, all fonts used in a document must be embedded; in order to guarantee exact color reproduction all colors must be specified in a device-independent way. **Metadata** must be embedded using the **XMP** format; encryption must not be used.

In addition to these straight-forward requirements, however, PDF/A requires various other PDF features which are more subtle (e.g. certain entries in font data structures), and prohibits some critical structures (e.g. certain combinations of TrueType fonts and encodings). There are many aspects which must be implemented and checked by software developers before they arrive at fully standard-conforming PDF/A products. PDF/A is much more than simply »PDF with embedded fonts«!

**PDF/A support in the PDFlib product family.** PDFlib provides application developers with a toolkit which allows the following PDF/A-related operations:

- ▶ Create PDF/A from scratch, e.g. based on text from a database
- ▶ Convert raster images (e.g. scans) to PDF/A
- ▶ Process existing PDF/A documents, e.g. merge or split
- ▶ Create PDF/A-1a with structure information (Tagged PDF)
- ▶ Attach **XMP metadata** to the generated documents, including the subtle topic of **XMP** extension schemas (see below).

All of these operations can be implemented with simple PDFlib function calls. Sample code for a variety of programming languages and development environ-

ments is provided with the [PDFlib](#) distribution. Additional programming techniques for PDF/A are available in the [PDFlib Cookbook](#). Due to the significant overlap between PDF/A and the PDF/X standard (ISO 15930) for the graphics arts industry, the PDF/A support in [PDFlib](#) took advantage of the fact that we've been supporting various flavors of PDF/X for several years. To facilitate font embedding as required by PDF/A, the Japanese Resource Kit for the [PDFlib](#) Family offers common Japanese fonts with a license for embedding, as well as country-specific ICC profiles, CMaps, and documentation for Japanese users.

**Creating PDF/A-conforming output.** Creating PDF/A-conforming output with [PDFlib](#) is achieved by the following means:

- ▶ [PDFlib](#) automatically takes care of several formal settings for PDF/A, such as PDF version number and required [XMP](#) identification entries.
- ▶ The [PDFlib](#) client program must explicitly use certain function calls and options (e.g. for font embedding).
- ▶ The [PDFlib](#) client program must refrain from using certain other function calls and option settings (e.g. encryption).

If the [PDFlib](#) client program obeys to these rules, valid PDF/A output is guaranteed. If [PDFlib](#) detects a violation of the PDF/A creation rules it will throw an exception which must be handled by the application. No PDF output will be created in case of an error; there is no danger of creating non-conforming output if an error occurs. Details of required and prohibited operations are discussed in the [PDFlib](#) documentation.

**Device-independent color specification.** In order to maintain consistent color reproduction, PDF/A requires the use of device-independent color, usually achieved via ICC profiles or CIE Lab color specifications. The optional output intent describes the color characteristics of the document. While these concepts are widely used in the graphics arts industry, enterprise PDF developers are not necessarily familiar with color management. In this situation [PDFlib](#) makes it easy to create device-independent output regardless of the source of input data:

- ▶ PDF/A output can be created with or without an ICC profile in the output intent.
- ▶ In the common case of black text [PDFlib](#) will automatically choose an appropriate color space (Lab or DeviceGray) depending on whether or not an ICC profile for the output intent has been specified.
- ▶ External ICC profiles and profiles embedded in images allow fine-grain color control.
- ▶ ICC profiles for common application scenarios are provided with the [PDFlib](#) distribution so that valid PDF/A output can be achieved quickly.

Raster images, e.g. TIFF and JPEG, play a vital role in document creation. Scanned paper documents and photographs from digital cameras are common examples of raster image data in document workflows. While in modern workflows raster image data may already be device-independent (usually by means of an embedded ICC color profile) and thus are PDF/A-compatible, legacy image data will in many cases be device-dependent, such as black-and-white or RGB scans without any associated ICC profile. [PDFlib](#) supports both situations:

- ▶ ICC profiles embedded in raster image files will be honored.
- ▶ External ICC profiles can be applied to an image.
- ▶ As a fallback solution for legacy data of unknown origin [PDFlib](#) contains a built-in sRGB profile which matches many hardware and software environments.
- ▶ By specifying an ICC profile for the output intent, device-dependent image data can be used without applying an ICC profile to the images.

The [PDFlib](#) documentation discusses PDF/A color strategies for common application scenarios.

**XMP made easy.** PDF/A mandates the use of **XMP metadata** for storing information about a document inside the PDF itself. **XMP** provides a powerful and flexible framework for storing standard and custom **metadata** (see also our separate whitepaper on **XMP**). If you already use **XMP metadata** in your workflow, you can create full **XMP** streams which **PDFlib** will integrate into the PDF/A output. However, developers who are not familiar with **XMP** are not required to delve into this vast topic. **PDFlib** will create the **XMP** output required for PDF/A, and will automatically map classic document info fields to the corresponding **XMP** constructs as mandated by the standard. As a result, developers who wish to do so can leverage the power of **XMP**, while **PDFlib**'s automatic **XMP** generation will be adequate in situations with simpler **metadata** requirements.

**XMP extension schemas.** **XMP** is by its very nature extensible, i.e. company- or industry-specific **metadata** requirements can be met by constructing **XMP** extension schemas. PDF/A supports this concept, but for ease of future retrieval mandates that a machine-readable description of the schema must be embedded in the document according to certain rules. **PDFlib** products support **XMP** extension schemas for PDF/A (actually, **PDFlib** 7.0.3 was the first product with support for extension schemas). **PDFlib** validates user-supplied **XMP metadata** including extension schemas in order to guarantee that the generated output fully conforms to PDF/A.

More details on **XMP** in PDF/A, plus an online validator for **XMP** extension schemas can be found on [www.pdflib.com](http://www.pdflib.com).

**Processing existing PDF/A documents.** Additional rules apply when importing pages from existing PDF/A-conforming documents. For example, in Adobe Acrobat it is very easy to combine two PDF/A documents such that the resulting output document no longer conforms to PDF/A (without any warning). When dealing with existing PDF/A documents, **PDFlib**+PDI carefully examines the PDF/A properties of all input and output documents to make sure that the output will again conform to PDF/A. For additional control the output intent of an imported document can be copied to the output PDF, effectively cloning the PDF/A color properties of an existing document.

**Creating PDF/A-1a with Tagged PDF.** PDF/A-1a can be regarded as PDF/A-1b plus Tagged PDF: it requires structure information for the document and imposes certain conditions on fonts to make sure that the text can be properly interpreted. As a result, PDF/A-1a documents are fully accessible for users with physical disabilities. In addition to the visual appearance they also preserve the meaning of its contents.

**PDFlib**'s support for PDF/A-1a is based on the features for producing Tagged PDF: each content item can be placed at a particular location in the document's structure tree; content items which are not relevant for the document structure (e.g. headers and footers, pagination) can be tagged as artifacts which means that they will be ignored when repurposing the document (e.g. when the document is read aloud by software, or converted to some other format). Alternative text can be attached to images; the text can be read to visually challenged users by Acrobat.

Note that you will need detailed knowledge about the document's logical structure in order to create Tagged PDF. **PDFlib** will take care of the PDF-related details, but it cannot infer the document structure from its contents.

Tagged PDF support has been introduced in **PDFlib** 6. **PDFlib** 7 adds the capability to include annotations in the document's structure tree which improves the accessibility of links and other interactive elements.

Based on the existing Tagged PDF support, **PDFlib** 7 can produce output which conforms to PDF/A-1a. This makes **PDFlib** the first tool to support this advanced PDF/A level.

**Validating PDF/A.** When implementing a standards-based workflow it is good practice to deploy tools for validating the results against the standard. With respect to the PDF/A standard there are software tools available which check whether or not a given PDF document complies with the ISO standard.

PDF/A created with **PDFlib** fully conforms to the validation rules of the Preflight tool in Adobe Acrobat 9 which is one of the strictest PDF/A validators available on the market. Note that PDF/A validation in Acrobat 8 did not fully implement the ISO standard.

**PDFlib** GmbH is actively working with vendors of PDF/A validation tools to make sure that creators and validators have a common understanding and interpretation of the PDF/A standard.

At this time, **PDFlib** GmbH does not offer any products for validating PDF/A documents (i.e., check whether a document conforms to the standard) or for converting arbitrary PDF documents to PDF/A.

**PDF/A-conforming document processing with PDFlib PLOP.** The products **PDFlib** PLOP and PLOP DS offer several PDF processing features, all of which are PDF/A-aware. This means that processing PDF/A input files will either result in PDF/A-conforming output if the processing can be accomplished in a standard-conforming manner, or an error message otherwise. Some examples:

- ▶ Trying to encrypt PDF/A documents will result in an error message since encryption is not allowed in PDF/A. You must explicitly sacrifice the PDF/A status in order to encrypt a PDF/A document with PLOP.
- ▶ Using PLOP DS to apply a digital signature to a PDF/A document will create the signature field in a way which conforms to the PDF/A standard.
- ▶ You can use PLOP to inject **XMP metadata** in existing PDF/A documents. Since PLOP 3.1 supports **XMP** extension schemas for PDF/A it can be used as a post-processing step for existing documents, or to leverage **XMP** extension schemas even if the software for creating the documents in the first place does not support it.

**PDF/A Competence Center.** **PDFlib** GmbH is one of the founding members of the PDF/A competence center which strives to increase industry awareness of PDF/A and achieve cross-vendor compatibility. We are actively involved in the Technical Working Group (TWG) which publishes TechNotes on PDF/A-related topics. The TWG also published the Isartor test suite for PDF/A-1, a set of test files which can be used to rigorously check the standard conformance and coverage of PDF/A validation software. See [www.pdfa.org](http://www.pdfa.org) for more information.

**PDFlib** GmbH  
Franziska-Bilek-Weg 9  
80339 München, Germany  
phone +49 • 89 • 452 33 84-0  
[info@pdflib.com](mailto:info@pdflib.com)  
[www.pdflib.com](http://www.pdflib.com)

