PDF/A Competence Center

PDF/A – A Look at the Technical Side

The long-term archiving format PDF/A is a relatively new standard that opens up new possibilities for many industry sectors and users with regard to enabling digital documents to be read and processed even in years to come. This document focuses on some of the technical aspects of PDF/A.

The standard: ISO 19005-1 (PDF/A-1)

As the name indicates, PDF/A-1 is the first part of the ISO range of standards for PDF/A. This international standard was formulated by the ISO Technical Committee TC 171 SC2 WG5 and was published on October 1st 2005. The Technical Corrigendum of 2007 further improved ISO 19005-1. PDF/A-1a is based on PDF 1.4 (the PDF version introduced with Acrobat 5).

PDF/A is not a 'newly invented' PDF format – instead, a PDF/A file is a completely normal PDF file that is tailored in line with minimum requirements and prescribed and prohibited features. Traditional PDF does not always have to be completely unambiguous, but PDF/A has the clear aim of ensuring that the display of documents is entirely clear – both today and in the future.

PDF was invented by Adobe Systems, who specifically released the format for standardization.

PDF/A – Important aims and rules

What does PDF/A aim to achieve? PDF/A aims to produce files with static content that can therefore be visually reproduced completely precisely today and in many years time. Files that are subject to long-term archiving should work regardless of the device or operating system used. The future usability of PDF/A files must also be guaranteed in a manufacturer-independent manner – and this includes Adobe. PDF/A is a 'complete' format. This means that PDF files that comply with the PDF/A standard are complete in themselves and use no external references or non-PDF data. The PDF/A-1 standard is based on PDF/A specification 1.4, which means that it works within the technical scope of the functions available in Acrobat 5.

A range of rules must be observed when generating PDF/A files in order to meet the goals named above. For example, when generating PDF/A, it is im-

portant to embed all fonts and clearly specify all colors. Forms, comments, and notes are only permitted to a limited extent. Compression is allowed as a general rule, but LZW and JPEG2000 are excluded. Transparent objects and layers (*Optional Content Groups*) are not permitted. PDF/A uses rules for metadata that are based on XMP (*Extensible Metadata Platform*). Finally, a PDF/A file must identify itself as such.

PDF/A-1 levels: a and b (accessible and basic)

There are two PDF/A-1 levels: PDF/A-1a and PDF/A-1b. These two conformance levels allow for the fact that different user groups have different requirements of a file format for long-term archiving and that the source material can vary greatly.

- **PDF/A-1b (basic)** is easier to generate and guarantees the visual reproducibility of content.
- **PDF/A-1a (accessible)** generally requires more effort during the generation process but offers enhanced access to content (especially for text). The structure of PDF/A-1a has the advantage that the reuse of text – for example, by means of an export function – is generally unproblematic, avoiding issues such as misaligned paragraphs and truncated words. PDF/A-1a also brings benefits for search functions. (To avoid any misunderstanding, please note that PDF/A-1b files also provide searchable text.) Lastly, users who want to create accessible PDF documents have – in PDF/A-1a – a format that is optimally suited to their needs.

What does PDF/A require to be PDF/A-1a-compliant?

PDF files that comply with the PDF/A-1a standard must fulfill additional prerequisites that enable PDF/A-1a to offer the benefits outlined above:

To enable precise searchability, all text must be reproducible using Unicode. Unicode is a system that enables characters (letters, digits, and symbols in international and historical font systems) to be precisely mapped to a code.

- The structure of a document is obtained using *tagged PDF* in PDF-1a.
- *Tagged PDF* supports the generation of accessible documents that can be read out by software such as the Adobe Reader and Acrobat screen reader functions.
- The document structure also enables the improved conversion of PDF files to other formats.
- A further advantage is the text reflow function (reformatting of pages to adjust them in line with the monitor/window). This permits PDF files to be read comfortably even on small devices (handhelds and cell phones).
- Finally, the preservation of a document's semantics is important for long-term archiving.

PDF/A-2 will follow on from PDF/A-1. What will this actually mean?

The publication of PDF/A-2 is planned for the end of 2008/start of 2009. This PDF/A standard will enable transparency and more recently implemented PDF features, since it is based on PDF 1.7 (PDF/A-1 is based on PDF 1.4). These new features include the following: JPEG2000, PDF layers (*Optional Content Groups*), *UserUnits* (page scaling), new comment types implemented since PDF 1.4, and Unicode paths for hyperlinks.

PDF/A-2 will also apply stricter rules for glyphs in embedded fonts. PDF/A-2 will not allow the use of *.notdef glyphs* and will only permit so-called 'empty' glyphs for *white space.*

The following three conformance levels will exist for PDF/A-2:

- **a:** As in PDF/A-1 (a = accessible)
- **b:** As in PDF/A-1 (b = basic)
- **u:** New – text can be mapped to Unicode (*u* = Unicode)

Documents that are stored in PDF/A-1 format today will remain valid following the introduction of PDF/A-2. However, future PDF/A versions will not always be backwards-compatible.

PDF/A: Practical questions

For PDF/A, some PDF areas have to fulfill certain prerequisites so that they can be unambiguously reproduced and therefore be considered to be future-proof.

Text

Text is displayed using fonts. PDF/A outlines certain rules for fonts in order to enable the precise reproduction of content today and for a long time in the future.

First, it is important to ensure that all fonts used in a PDF file are embedded. All glyphs used must be stored within the PDF itself – a simple reference ('load font xyz here') is not sufficient for PDF/A. Another requirement is that the character set encoding must be achieved in a way that enables the intended depiction of text. If, for example, the tracking is incorrect, a PDF file cannot be precisely reproduced and is therefore not PDF/A-compliant.

Problems with glyphs

Glyphs are graphical representations of characters (letters, digits, and symbols). All of us have seen cases where problems have occurred when trying to display font characters. For example, characters that are completely missing result from an incorrect sub-setting for TrueType fonts – only an 'empty' glyph is displayed in this case.

The missing 'ä' is caused by an incorrect sub-setting for TrueType fonts.

In the case of Type 1 fonts, however, the system uses a *notdef glyph* (replacement character) instead. This is often simply a box containing an 'X', as shown in the graphic below.

An 'X' instead of a gap: Type 1 fonts fall back on a replacement character for undefined characters.

The inappropriate multiple usage of subset fonts is generally an effect that can occur as a result of problems when generating a PDF.

Bugs in the viewer or printer can also result in incorrect reproductions – the incorrect caching of a font instance can have negative effects.

- If there are different subsets for a single font, the first subset is used for all other subsets.
- Identical fonts are used with different encoding.
- Different subsets have the same 'unique' name and the same base font.
- Remedy: Each font in a PDF has a unique name (also applies to the base font).

Encoding – incorrectly correct?

Problems can occur before and/or during the generation of PDFs. A PDF/A file can be formally correct yet still have incorrect glyphs. Only a careful visual check can uncover this problem. Because generation problems also affect Unicode mapping, the problem attracts the attention when a visual check is carried out on the extracted text.

In PDF/A, text/font usage is specified uniquely enough to ensure that it cannot be incorrect.

If viewers or printers do not offer complete support for encoding systems, this can result in problems with regard to PDF/A.

Glyph width: PDF versus font

Inconsistencies can occur for glyph width specifications. To ensure that this does not occur, the specifications in the PDF Font Dictionary must correspond to the specifications in the embedded font. In reality, slight deviations resulting from differing dimensioning are unavoidable and must be tolerated. Specifica-

tions in the *Width Array* property in PDF are usually integer values, but this is not mandatory.

PDF generation: Problems occur when the tracking specifications are based on a font other than the font that is actually embedded. The subsequent embedding of fonts therefore also involves risks.

Displaying/printing PDF/A: Problems can occur when displaying or printing PDF if the viewer or printer being used uses a replacement font rather than the embedded font.

The quick brown fox jumps
over the lazy dog

The qui ck brown fox j umps
overthe l azy dog

If the tracking specifications do not match, the text cannot be displayed properly.

Clearly defined colors

Incorrect colors can result in a completely different message being imparted by an image than was originally intended. It is therefore important for colors to be reliably reproducible in PDF/A. In relation to this task, color profiles act as 'instruction leaflets' that give information on how to handle colors.

Colors can impart emotion – the correct display of colors can be a decisive factor in ensuring that the intended message of an image is correctly interpreted.

The precise display of colors via color management is important not only for photographs in PDF/A but also for fonts or graphics – just think, for example, of the corporate image of a company.

PDF/A provides several ways of ensuring the exact reproduction of colors.

- PDF/A can use source profiles (or CalGray/CalRGB/Lab) or a target profile in the OutputIntent property (output condition). If all color spaces are qualified with a source profile (or via CalGray/CalRGB/Lab), no output intent is required.
- Device-independent grayscale (DeviceGray) is 'covered' using a target profile, either grayscale, RGB, or CMYK.
- If both device-independent RGB (DeviceRGB) and device-independent CMYK (DeviceCMYK) occur in a document, only one of them can be covered by the output intent.

If using CMYK, it is important to remember that the ICC profiles can be extremely large – in particular, 'prtr' profiles (output profiles for printers) can require memory space of between 500 KB and 2 MB.

Note: For Separation/DeviceN color spaces: The so-called *alternate space* is not subject to the same requirements as the process color spaces.

Comments/annotations

Some comments are allowed, but other comment types are prohibited. Comments in the form of movies, audio clips, and attachments are not permitted (additional programs are required to display these types of comment, and the programs in question may not be available in the future). In addition, any feature that was implemented after PDF 1.4 is not PDF/A-compatible. This includes *Polygon, PolyLine, Caret, Screen, Watermark,* and *3D. Highlight markup annotation* is permitted but not always desirable, since this type of annotation usually uses transparency, which is not permitted in PDF/A-1.

With regard to hyperlinks, the special 'Link' annotation type is permitted with or without activated 'Appearance'. As a rule, URL links and other links are permitted – they must be 'somehow' displayed by a PDF/A viewer, but they must not necessarily be executed. As far as the standard is concerned, it does not matter whether or not a link references a valid target.

Form fields: Permitted but with restrictions.

As a general rule, all form fields are allowed. They may be linked to certain actions, but may never have JavaScript. There may be only one appearance in order to ensure that there can be no rollover effects. PDF/A-suitable form elements may not be hidden – they must be visible and set to print.

If you still wish to convert a form with critical functions into a PDF/A file, you can 'flatten' its form fields. While reducing/flattening form fields achieves the required PDF/A-1 conformity, the assignment of content to the form fields in question is lost. As an alternative, you can avoid problematic components in advance with the following measures:

- Remove certain actions; always remove JavaScript
- Remove all *appearances* except for the *default appearance* (decide on a single depiction)
- Remove all hidden fields
- Set the entire form to visible and to print

Actions and JavaScript

The following actions are explicitly forbidden:

- *Launch, Sound, Movie, ResetForm, ImportData,* and *JavaScript*
- *set-state* (no longer used in some recent PDF versions) and *no-op* (was described in PDF 1.2 but has not been implemented
- *Named actions* other than *NextPage, PrevPage, FirstPage,* and *LastPage.*
- JavaScript and *additional actions (AAs)* are always prohibited.

Implicitly forbidden actions (actions that are forbidden because they were implemented after PDF 1.4):

- *SetOCGState* (set status for *Optional Content Groups* of PDF layers), *Rendition* (controls the play-back of multimedia content), *Trans* (page transition), and *GoTo3DView*

Another action that is implicitly forbidden is:

- *Hide* (hide comments – the execution of this action would make the PDF/A file invalid)

Some actions are allowed:

- *GoTo* (go to a destination), *GoToR* (go to remote, go to a destination in another document), *GoToE* (go to embedded, go to a destination in an embedded file), *Thread* (go to a certain article thread), *URI* (Uniform Resource Identifier), and *SubmitForm*
- In addition, the following *named actions* are permitted, as mentioned above: *NextPage, PrevPage, FirstPage,* and *LastPage.*

Metadata (at document level)

The only absolutely necessary metadata specification is the PDF/A-1 property along with the relevant conformance level (PDF/A-1a or PDF/A-1b) in 'Metadata' (XMP). All other metadata is optional.

If using schemata that are not predefined in the XMP specification, the schema in question must be embedded in the XMP.

If traditional document properties exist, they must be mirrored in the XMP metadata in order to achieve PDF/A conformity. This applies to the following: *Title*, *Author*, *Subject*, *Keywords*, *Creator* (application), *Producer* (PDF created with…), *CreationDate,* and *ModDate*.

The namespace and metadata mapping sections of the standard are formulated in a way that is not completely clear. For a better explanation, see the Technical Corrigendum (cf. TechNotes 0001 and 0003 at: http://www.pdfa.org/doku.php?id=pdfa:en:techdoc).

Olaf Drümmer, callas software/aoe