PDF 2.0

what does it mean for us

context **2024** meeting

Short history

- It started in 1992.
- The pdf format basically is flattened PostScript.
- The file has objects accessed via a page tree.
- Random access is provided via a cross reference table.
- In principle not all in the file has to be loaded.
- All resources (fonts, graphics) are (can be) included.

The tools

- A pdf file was supposed to be made with Acrobat Distiller.
- Previewing was done with Acrobat Reader.
- Distribution was assumed to happen with Acrobat Exchange.
- The Reader was limited in functionality.
- Exchange was expensive (pay per document or cd).
- Distiller didn't come cheap either.
- Plugins were part of the design but in the end a failure.
- The msdos reader was not that bad.
- But such a lightweight viewers never made it to e.g. linux

Usage

- For printing it was to replace PostScript: high speed and less demanding.
- It was suitable for for preflight and last minute fixes: object version numbers and appended objects.
- It could be used for storing graphic editor states: undocumented extensions are possible (using objects).
- At some point widgets (forms) entered the picture (but intended usage changed, e.g. fdf).
- Layers are a powerful feature.

Usability

- Media support is unreliable and changed: quicktime, flash, whatever, a missed opportunity
- Widgets are a bit unreliable, especially initialization and inheritance (bugs becoming features).
- There is no baseline JavaScript defined so viewers lack some simple powerful things.
- One can do a lot but open source viewers (always) lag(ged) behind.
- Some bits (like tagging) have seen little use and support so one can wonder where that ends.
- For instance layers could be more useful but they lack control and support in free viewers.

Standardization

- In order to be predictable we have all kind of pdf standards (prepress, archiving, accessibility).
- The main (big) standard is an iso specification (kind of semi free).
- Version 1.7 was already more or less frozen for a while.
- So version 2.0 is not really a big jump, it is mostly 1.7, so maybe more of a freeze.
- Validation and preflight is big business . . .
- . . . as is signing and digital right management.
- The 2.0 standard seems kind of fluid anyway (also driven by what tools can(not) do).
- Printing houses often have old tools (and sometimes mess with the pdf).

Producing pdf

- The tools produce quite reliable and compact pdf.
- We can basically add anything we want.
- There is no real need to adapt as pdf 2.0 it is.
- Type3 support in open source is inconsistent and needs care.
- There is demand for tagging (weak and insufficient spec).
- By going 2.0 we can in principle drop older versions.
- We do our best to deal with encryption and signing.

Open source

- TEX engines could produce pdf (we used dvipsone and Distiller) rather early in the game.
- Interactive features could be supported immediately (we already supported dviwindo) because we had an backend abstraction layer.
- The real take off happened when pdfTEX came around.
- An reliable alternative route was via dvipdfmx.
- For us Sumatra (MS Windows) was the first competing alternative viewer.
- And Okular (linux, MS Windows) was quite useful too.
- I now use these two and seldom launch Acrobat Reader which has rather intrusive interfaces.
- It is very much about not letting the tools getting in the way (productivity).

Closed source

- There is a lot of commercialization of pdf.
- (Okay, that also happens in the TeX environment.)
- We have and get locked-in dependencies on the cloud and web.
- No one really knows what happens with data and content.
- One can be surprised about the messy pdf being produced.
- Standards become applications, applications become standards.
- Open and closed source are different worlds with often different interests.

New in ConTeXt

Inclusion of pages from a pdf file is controlled by:

```
compactors-preset.lua
```

We can enable and disable checks and fixes if needed. If needed we can add some hooks.

There are also options to merge references, comments, bookmarks, fields, layers and renditions.

In LMTX we add additional information that we can use in the future. We have a registered name space (in addition to the standard one used in the other TeX engines).

Demo: fixing and normalizing rather hybrid documents.

Discussion: What do users need?