PDF libraries and TEX

Martin Schröder

EuroTEX 2009
31st August – 4th September 2009
Den Haag

Why this talk?

- Over the last years a number of new PDF libraries have appeared
- We now have three free TEX engines that can read and write PDFs: pdfTEX, luaTEX, XƎTEX
- Ideally these engines would use one (maybe the same) well designed and cleanly written library for reading and writing PDF – currently they don't. So should they switch to one of the existing libraries?
- Or maybe you want to write a program that handles PDF and are looking for a library

What is in a PDF library?

- PDF is a relatively complex file format with a lot of different object types
- Most PDF libraries are designed for *creating* PDF
- Only a handfull of PDF libraries support *reading* PDF
- Very few PDF libraries are designed for *modifying* PDFs

What to look for in a PDF library

- Programming language
- License (BSD or GPL)
- Actively maintained
- Quality of documentation
- Level of abstraction – does it only know about the basic object types or can you ask it for the number of visible layers on page 7?
- Reading and writing; incremental writing (modifying)
- PDF 1.5 (compressed object streams)
- Fonts (OTF?) and colours
- Large File Support (LFS) (files >4 GiB)
- Parsing of content streams
- Support of XMPP
- Unicode?

What does a TeX engine need from a PDF library?

- Support for writing PDFs: Create a PDF, create pages, place text on a page (with absolute positions and kerning etc.), switch fonts and colours, handle font embedding and subsetting, place images, set links, set meta information, set other PDF structures (annotations, layers. . .), embed literal PDF code.
  Ideally we'd have a high-level interface, but now this is mostly handled in a non-abstract way in the engine code.
- Support for reading PDFs and getting information about PDFs: Size, number of pages, fonts, colours, meta information, layers, images. . .
  Now the engines use library code for this where possible, but the library we use (poppler/XPDF) doesn't offer everything we need, so we also have to use the low-level interfaces (e. g. parse the dictionaries ourself).

Why should a TEX engine use a PDF library?

- Using an existing library would free the developers from having to handle PDF features themselves and would get us (hopefully) well-supported code used by others
- It would expand our possibilities for reading (and writing) PDF
- If it would use an abstract interface for the engine, other output formats could be provided by a different library

pdfTEX

- pdfTEX uses XPDF for PDF inclusion
- XPDF is written in C++ and used only in one source file (`pdftoepdf.cc`) of pdfTEX (which is Pascal and C otherwise)
- There is no layer of abstraction between pdfTEX and XPDF
- XPDF is statically linked into pdfTEX
- Writing PDF is done without an abstract concept of PDF objects by pdfTEX itself
- Since TEXlive 2009 pdfTEX can use poppler instead of XPDF

luaTEX

- luaTEX is a child of pdfTEX: It also uses XPDF, and the PDF inclusion code is mostly unchanged. So is the PDF writing code, but a rewrite has started
- There is currently no layer of abstraction between luaTEX and XPDF
- XPDF is statically linked into luaTEX
- Since TEXlive 2009 luaTEX can use poppler instead of XPDF

XƎTEX

- XƎTEX uses XPDF to find the bounding box and orientation of included PDFs
- XPDF is statically linked into XƎTEX
- Since TEXlive 2009 XƎTEX can use poppler instead of XPDF
- xdvipdfmx has its own PDF parser written in C used for reading *and* writing

XPDF

- XPDF is a PDF viewer (and some command line tools) started in 1996 and written in C++
- Coding style feels like C(++), doesn't use newer C++ features
- Not designed as a library
- Dual-licensed: © Glyph & Cog, GPLv2 and commercial licenses are available
- Not much API documentation; no code documentation
- Medium level of abstraction
- Only support for reading PDFs; supports PDF 1.5
- No LFS; size of PDFs limited to <4 GiB
- No public source repository
- XPDF has a history of security problems (mostly buffer overflows)

poppler

- poppler is a fork of XPDF started in 2005 aimed at creating a free (GPLv2) PDF rendering library which is API-compatible to XPDF
- poppler's core can be easily substituted for XPF's code; indeed the XPDF viewer can be compiled with poppler as a backend
- poppler's main focus is rendering PDFs
- Not much API documentation; no code documentation
- Medium level of abstraction
- Only support for reading PDFs; supports PDF 1.5
- No LFS; size of PDFs limited to <4 GiB
- Uses git and make

podofo

- podofo is a PDF library (with reading and writing) started in 2006, written in C++ and licensed at GPLv2
- podofobrowser is a PDF object browser (using podofo and Qt) which can also rewrite PDFs
- Good API documentation, documented examples, some code documentation; documented coding style (modern C++)
- Aim is creating PDFs and some analysis; high level of abstraction for writing, medium level of abstraction for reading
- Fonts handled through fontconfig, initial work on font subsetting
- LFS
- Imposition tool which uses *Lua* for plan files
- Full unicode support on both Windows and Linux plattforms
- Initial work on content stream parsing
- Uses subversion and cmake

GNU PDF

- “The goal of the GNU PDF project is to develop and provide a free, high-quality, complete and portable set of libraries and programs to manage the PDF file format, and associated technologies.
  Right now the library is under heavy development and we have not released a version yet.”
- It’s written in C and (of course) licensed at GPLv3
- The project plan includes a full-fledged PDF viewer and editor called GNU Juggler
- The base layer has been mostly finished, the object layer is being designed
- Uses bzr and make
- Developement is slow

MuPDF

- MuPDF is a high quality PDF viewer started at Artifex (the company behind GhostScript) written in C and licensed at GPLv2
- Not much API documentation; no code documentation
- Very low level of abstraction
- No LFS; size of PDFs limited to <2 GiB
- Uses darc and perforce-jam

iText

- iText is a PDF library written in Java 1.4 initially aimed at writing (lately some reading and modifying has been added) licensed at MPL or LGPLv2; commercial licenses are available
- Documentation is also available as a book
- pdftk is a command line tool written in C using iText (thanks to gcj) which allows some manipulations of PDFs; it's mostly unmaintained (last release from November 2006)

jPod

- jPod is a free (BSD) Java library for reading and writing PDFs. It can handle content streams and has some quite advanced features
- jPodRenderer is a renderer based on jPod licensed at GPLv3

PDFlib

- PDFlib is commercial C library aimed at creating PDFs from web services; lately PDF import functions have been added.
- Bindings for C, C++, Java, Perl, PHP, Python, Ruby, TCL and REALbasic are available
- Runs on Unix, Mac and Windows
- Software available for automatically filling in templates (blocks) in PDFs
- There's also a free (own license) variant of the library from which pdfTEX borrowed some ideas for the handling of PNG files

Others I

- PDFBox is a free (BSD) Java library for reading and writing PDFs
- Apache FOP has a Java library for writing PDFs licensed at Apache License 2
- PDF Clown is a free (GPLv2 or LGPLv2) Java PDF library for creating and writing PDFs with multiple abstraction layers
- Big Faceless Java is a commercial PDF library for creating and writing PDFs
- Multivalent is a free (license unclear – GPLv2?) viewer written in Java for HTML, PDF, *DVI*, man pages, and other document formats; it supports reading, writing and modifying up to PDF 1.5. The latest release is from January 2006. Source is currently not available (despite the GPL) and there are some non-free tools developed with it available.

Others II

- PJX is a simple Java library supporting reading, writing and modifying licensed at GPLv2
- libHaru is a free (zlib) C library for generating PDFs
- jagPDF is a free (BSD) Java library for generating PDFs
- Adobe and Global Graphics sell commercial PDF libraries
- There are many abandoned or unfinished free PDF libraries (luaTEX?)

Inkscape

- Free (GPLv2) multi-plattform vector graphics program
- Can read and write PDF
- PDF parser uses code from poppler
- PDF writer uses Cairo

Scribus

- Free (GPLv2) multi-plattform DTP program
- Can write PDF and handles Type1, TrueType and OpenType fonts
- PDF writer is written in C++ and tailored for the documents created by Scribus (no general purpose library)
- Maintainer of podofo is a team member of Scribus

Open Office

- Free (GPLv3) multi-plattform office suite
- Can read and write PDF and handles Type1, TrueType fonts; no support for OpenType fonts
- PDF reader and writer are written in C++; not designed as a library

Conclusion

- There is no ideal free PDF library yet
- XPDF (in pdfTEX, luaTEX) is showing its age
- poppler is a ready substitute for XPDF (and already used)
- podofo is the future (for C++); let's extend it for the use in TEX engines