

pdfTeX 1.40: What's new

Martin Schröder
Crüsemannallee 3
28213 Bremen
Germany
martin@pdfTeX.org
<http://www.pdfTeX.org>

Abstract

The latest version of pdfTeX, 1.40, was released at the start of 2007. We will present its new features and have a look towards the future.

1 The past

pdfTeX turned 10 on 15th March (Hàn Thê Thành renamed T_EX2PDF to pdfTeX on the 15th March 1997), and we want to present the latest release. But let us first look back at our previous release, which was pdfTeX 1.30.0 on 1st August 2005. Since then there have been six intermediate releases fixing bugs (mainly security problems with XPDF); version 1.30.6 was released on 16th February 2006. Its main enhancements were improvements in the handling of PNGs (alpha channel and transparency, 16-bit colour and gamma correction), macros for timekeeping, random numbers, string conversions, and file functions. Also pdfxTeX was gone; all enhancements were now in pdfTeX and pdf ϵ -TeX.

2 The present — 1.40

After 17 months of development we released pdfTeX 1.40.0 on 1st January 2007. At the time of writing there have been five intermediate releases fixing bugs; version 1.40.5 was released on 31st July 2007. pdfTeX 1.40.x is included in T_EX Live 2007 and MiKTeX 2.6. The main internal change is that we merged all the sources (i. e. change files) for T_EX, ϵ -TeX and pdfTeX into the two files `pdftex.web` and `pdftex.ch` (for Web2C etc.). This makes maintaining the sources much simpler. Also pdf ϵ -TeX is gone as a separate program; pdfTeX now contains ϵ -TeX.

2.1 PDF

pdfTeX is now able to generate object streams, a feature of PDF since PDF 1.5 [1]. A PDF file consists of objects and a cross-reference table for fast access



This work is licensed under the Creative Commons Attribution-NonCommercial 2.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nd/2.0/de/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

to these objects. But formerly only the stream part of the objects could be compressed. So if a PDF had many non-stream objects, its size could not be reduced efficiently. Object streams are a kind of meta-objects; they can contain many (up to 256) non-stream objects. From pdfTeX 1.40 onward object streams can be compressed as a whole, which leads to smaller PDF files. The generation of object streams is controlled by the `\pdfobjcompresslevel` parameter (only with PDF ≥ 1.5):

- 0 The default; no object streams are generated.
- 1 Object streams are generated, but the Document Information Dictionary and included PDFs are not compressed.
- 2 The Document Information Dictionary is not compressed, everything else is.
- 3 Everything is compressed.

Another improvement leading to smaller PDFs is that pdfTeX now writes out the widths of the characters in the fonts (`/Widths`) with a higher precision and so rarely has to position characters separately. Previously this was done with a lower precision, leading to many adjustments of single character positions in the PDF.

The new primitive `\pdflastlink` now gives the object number of the last link created with `\pdfstartlink`.

PDF/X, the new ISO standard of PDF, requires the setting of `/ModDate` and `/Trapped` in the Document Information Dictionary. These keys have now default values that can be overridden with `\pdfinfo`.

Also the additional PDF statistics in the log file about the number of objects etc. are now correct; previously they were written out too early and thus missed the objects for e. g. embedded fonts.

2.2 JBIG2

pdfTeX can now handle another format of image files: JBIG2. JBIG2 is an image compression standard

for binary images, developed by the Joint Bilevel Image Experts Group (which is also responsible for the JPEG standard). It is suitable for both lossless and lossy compression. According to a press release from the Group, in its lossless mode JBIG2 typically generates files from 3 to 5 times smaller than Fax Group 4. PDF supports JBIG2 since PDF 1.4, but previously there were no free encoding programs for JBIG2, only decoding programs, so the OSS world was unable to generate JBIG2 files. This changed in 2006 when Google sponsored the development of a free encoding program (`jb2enc`). pdfTeX (and L^AT_EX with a recent `pdftex.def`) support JBIG2 files with `.jb2` or `.jbig2` suffixes.

2.3 Colour stacks

Colour in pdfL^AT_EX has an old problem: If you have different colours on the page and in the footnotes, you'll probably get the wrong colour after a page break. The `pdfcolmk` package tried to fix this, but it was a kludge. To fix this, pdfTeX 1.40 introduces support for colour stacks; L^AT_EX gets this through `pdftex.def` and some packages (e.g. `pdfcolfoot`). Colour stacks are handled with these commands:

- `\pdfcolorstackinit [page]` initializes a colour stack and expands to the number of the stack. With `page` you get a stack that is reset at the start of every page.
- `\pdfcolorstack <stack number> push {<colour>}` saves the `<colour>` on the stack and outputs it.
- `\pdfcolorstack <stack number> pop` removes the topmost colour from the stack and sets the now topmost value.
- `\pdfcolorstack <stack number> current` gets the topmost colour from the stack and sets it, but doesn't change the stack.
- `\pdfcolorstack <stack number> set {<colour>}` sets the topmost colour of the stack, but doesn't change the rest of the stack.

2.4 Transformation matrices

PDF (like PostScript) uses transformation matrices for positioning objects. Before pdfTeX 1.40, matrix changes were done and hidden inside `\pdfliteral` nodes, but pdfTeX doesn't parse the argument of `\pdfliteral` and so does not know the new settings of the transform matrix, which might conflict with pdfTeX's use of the matrix. pdfTeX 1.40 adds new primitives to save pdfTeX from parsing `\pdfliteral`'s argument and to notify pdfTeX about matrix changes for use in calculating link and anchor positions.

- `\pdfsetmatrix{<a> <c> <d>}` is the equivalent of `\pdfliteral{<a> <c> <d> 0 0 cm}`
- `\pdfsave` is the equivalent of `\pdfliteral{q}`
- `\pdfrestore` is the equivalent of `\pdfliteral{Q}`

Some remarks:

- T_EX already supports translations, thus the matrix is limited to four values, for scaling and rotating.
- There are some restrictions about `\pdfsave` and `\pdfrestore`:
 - They must be properly nested.
 - A pair must start and end in the same group at the same level.
 - A pair must start and end at the same position.

Happily these restrictions are satisfied by the `graphics` package.

The latest `pdftex.def` uses these primitives.

2.5 General enhancements

- pdfTeX now offers limited support for namespaces:
 - `\pdfprimitive<TEX-primitive>` executes the original `<TEX-primitive>`, even if its definition has changed. Thus

```
\let\relax\undefined
\pdfprimitive\relax
```

 works and doesn't give an error.
 - `\ifpdfprimitive<TEX-primitive>` is true if `<TEX-primitive>` still has its original meaning.
- `\ifpdfabsnum` and `\ifpdfabsdim` are like `\ifnum` and `\ifdim`, but don't care about signs.
- The memory areas for PDF objects (`obj_tab`) and names (`dest_names`) now grow dynamically as needed, making corresponding settings in `texmf.cnf` obsolete.
- `\pdfsavepos` now also works in DVI mode.
- The resolution of PK files is now read from `texmf.cnf` if it hasn't already been set in the format or document.
- In almost all cases of fatal pdfTeX errors (i.e. if the resulting PDF would have been invalid anyway) no PDF is generated.
- The format of warnings and error messages has been revised and unified.

- If called with `-version` pdfTeX now tells the versions of the libraries compiled with and actually used:

```
Compiled with libpng 1.2.15; using libpng 1.2.15
Compiled with zlib 1.2.3; using zlib 1.2.3
Compiled with xpdf version 3.01
```

- pdfTeX now can be switched into a draft mode with `-draftmode` and `\pdfdraftmode=1`. In draft mode pdfTeX does everything it normally does, but does not write a PDF and does not read the contents of included images, thus speeding up the execution. This is useful e.g. if you know you need another two L^AT_EX runs to get the references right.

2.6 Fonts and HZ

- pdfTeX now supports subfonts: All needed map entries are generated automatically together with the Unicode mappings.
- pdfTeX can generate ToUnicode entries for Type1 fonts with `\pdfgentounicode` and `\pdfglyphtounicode`.
- Previously with font expansion in autoexpand mode for every expansion a complete new font was included in the PDF. Now the font is only included once and gets expanded on the fly by using the text matrix. This leads to smaller PDFs and enables the use of HZ with TrueType fonts and even non-embedded fonts (e.g. Times-Roman).
- Hàn Thế Thành describes more improvements in his paper [2].

2.7 Shell escape

If the first character of a file name for `\openin`, `\openout` and `\input` is a “|” (and `\write18` has been enabled), the rest of the file name is executed as a command. Some examples:

```
\openin1= "|ls -l"
\loop \unless \ifeof1
  \read1 to \cs \message{\meaning\cs}
\repeat
```

outputs the filenames in the current directory.

```
\openout1= "|sort >alphabet.tex"
\write1{b}
\write1{a}
\write1{c}
\closeout1
```

generates a sorted file.

The shell escape feature is available in all Web2C-based T_EX engines, e.g. X_TT_EX and pdfT_EX.

3 The future

The future of pdfT_EX is luaT_EX: The pdfT_EX team will take over the maintenance and development of luaT_EX once its initial development has been finished. This will offer support for Unicode and OpenType and integrate Lua [3], thus finally giving T_EX a proper programming language. pdfT_EX will still be maintained for those needing a time-proven engine, but new features will be added only to luaT_EX.

References

- [1] Adobe Systems Incorporated. *PDF Reference*. Sixth edition, 2006.
- [2] Hàn Thế Thành. Font-specific issues in pdfT_EX. In *Proceedings of the EuroT_EX 2007 conference*, 2007.
- [3] R. Ierusalimschy, L. H. de Figueiredo, W. Celes. *Lua 5.1 Reference Manual*. 2006.