

T_EX and prepress

Abstract

This article discusses preparing documents for professional printing with T_EX and pdftex, including color printing and prepress standards.

Keywords

PDF, PostScript, color, Acrobat, separations, overprint

History

Most of us aren't graphics professionals. Still, now and then we have things that need to be printed professionally at a conventional printshop.

A bit of historical perspective: originally, we dealt with this by supplying 'camera-ready' laserprinter output to the printshop, from which printplates were created photographically. This method certainly prevented surprises, but was not the way to get quality output.

During the nineties, PostScript dumps became increasingly popular among T_EX users as an alternative. Professional-quality output became a real possibility. But it might take some effort to find a printshop willing to process raw PostScript. The usual practice in the graphics industry was handing off application files. Of course, this had its drawbacks: it was easy to forget to include a font or a graphic file in the job, and the printshop from its side had to watch against reflow, *i.e.* changes in linebreaks. For T_EX users, this practice was no option at all.

T_EX users have for a long time been using Ghostscript for previewing, converting and printing PostScript. However, most printshops seem to have been unaware of such tools. And without such tools, a PostScript file is pretty much a black box.

Then Adobe developed PDF, a derivative of PostScript, and has had some success in persuading the graphics industry that a PDF-based 'workflow' is the way to go. By now, it is not that hard any more to find printshops accepting jobs in PDF format.

PDF tools

PDF has been developed both as a more tractable format for print production and as a format for various interactive uses. Whereas PostScript is a full-fledged programming language, PDF lacks programming features. Presumably, this made it easier to write software

for it, and we certainly have seen a flood of software for PDF. Just pay a visit to www.planetpdf.com to convince yourself.

These include of course the Adobe Acrobat programs: the free Reader (which is now named Adobe Reader) and the various commercial editions of Acrobat. All these commercial editions include Distiller for converting PostScript to PDF. As of this writing, the latest versions (6.xx) of the Reader and the other Acrobat programs are available only for Windows and Mac OS X.

Other PDF tools include various third-party Acrobat plugins, for prepress functions such as color separation and page imposition, and for limited editing. Also toolkits/libraries for programmers, some of them open source. There are also commercial and free alternative PostScript-to-PDF converters, Ghostscript not the least among them. Mac OS X Panther contains a command-line utility `pstopdf` which is quite good. Many programs now can generate PDF directly.

The principal open source PDF readers are Ghostscript (via a suitable frontend such as `gv` or `GSView`), and `xpdf`. The latter is part of a suite. `xpdf` itself requires X11, but the rest of the suite consists of some very useful command-line utilities which are also available for Win32. I'll mention some of them below.

Routes to PDF

The principal routes to generate PDF from T_EX are:

- from T_EX to dvi to PostScript, and then running the PostScript file through Distiller or another PostScript-to-PDF converter
- from T_EX directly to PDF, using `pdf[e]tex`
- from T_EX to dvi and then with `dvipdfm[x]` to pdf. `Dvipdfm-cjk`, a.k.a `dvipdfmx`, offers extended support for CJK (Chinese/Japanese/Korean) languages with their huge character sets.

One reason for choosing the roundabout way via PostScript is when you use PostScript-specific features such as the `pstricks` package which haven't been adapted to PDF. Another reason is that you may need Distiller's extra prepress-related controls.

If you need `pdftex`-specific features but also Distiller's controls, then you can go from PDF to PostScript,

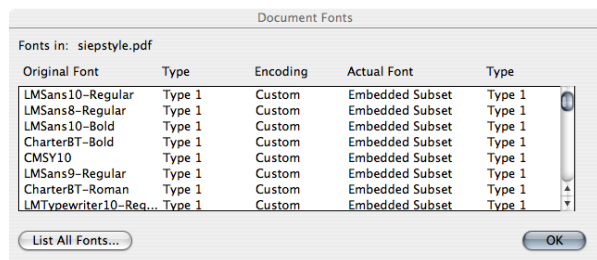


Figure 1. Adobe Reader: Document Fonts

and then back to PDF. For the first conversion, you can use either Adobe Reader or Ghostscript or pdftops (from the xpdf tools suite), for the second one either Distiller or one of its alternatives. This usually works just fine.

Ghostscript as a PDF generator. Many of Distiller's prepress-related controls are also available via Ghostscript; a fairly thorough description can be found in the ps2pdf manual that is included in the Ghostscript distribution.

Preventing font problems

Acrobat used to come with a base set of fonts: Courier, Helvetica, Times, Symbol and Zapf Dingbats. Therefore, these fonts were customarily not embedded. To the dismay of the T_EX community, in Acrobat 4 Times was replaced with Times New Roman, and Helvetica with Arial. Grudgingly, we concluded that it was better to avoid ambiguity and embed *all* fonts for print, including the base 14, and just put up with the increase in file size. Fortunately, this version of Acrobat also introduced joboptions files, which were named sets of Distiller settings. This made it easier to switch between generating unambiguous pdfs for prepress and small pdfs for online viewing, where you may prefer to exclude the base-14 fonts.

Another point of concern is MetaFont-generated bit-mapped fonts. Although these may look fine in print, they look pretty bad on screen, and PDF validation tools will probably flag them as undesirable or illegal.

Font embedding is controlled by map files. For teT_EX/fpT_EX/T_EX Live, these used to be located under texmf/dvips/ and texmf/pdftex/, but are being relocated to texmf/fonts/map/engine, engine being e.g. dvips or pdftex. Make sure that the relevant mapfiles contain entries for the Computer Modern fonts, and that all entries contain a font filename:

```
ptmr8r NimbusRomNo9L-Regu
" TeXBase1Encoding ReEncodeFont "
<8r.enc <utmr8a.pfb
```

(a single line), rather than

```
ptmr8r Times-Roman
" TeXBase1Encoding ReEncodeFont " <8r.enc
```

The first version downloads the URW Times clone included in most free T_EX distributions, the second references a version of Times which should be available to either Acrobat or the printer/typesetter.

With the 2003 editions of teT_EX/fpT_EX/T_EX Live, maps are generated with a utility updmap, and configured either by editing web2c/updmap.cfg or with command-line parameters. Also check texmf/pdftex/config/pdftex.cfg to see which mapfiles are used by pdftex.

Changes are planned for future releases, so check the documentation if things don't work out.

As to MikTeX: The manual mentions updmap.cfg for manual configuration and the command initexmf --mkmaps for forcing regeneration of the mapfiles.

You can check your fonts with the Reader by *first scrolling through the entire document* and then either click File/Document Properties/Fonts... or by clicking the right-pointing arrow above the vertical scrollbar and select Document Fonts...; see figure 1.

If Acrobat doesn't support your platform, then use pdffonts from the xpdf suite instead:

```
> pdffonts siepstyle.pdf
name                type    emb sub uni object ID
-----
GZLRCN+LMSans8-Regular Type 1 yes yes no      10  0
EQOQAE+LMSans10-Bold  Type 1 yes yes no      13  0
...

```

Preventing problems with figures

Included figures also may cause problems:

Fonts: keep in mind that included pdfs may also contain fonts and font problems. If a font is embedded in a pdf that you are trying to include, and pdftex complains that it can't find the font, it may be that the font is present in the mapfile but absent from your installation. In that case, create a custom version of the mapfile without the entry. This will hopefully no longer be a problem with version 1.20.

Lines with width 0, as produced by several graphics programs when you select 'hairline'. Width 0 means one pixel wide. This looks fine with 300dpi output from a desktop printer, but becomes completely invisible with high-resolution typesetter output. A width of 0.3pt should be safe.

Resolution of pixel-based images. With the wrong Distiller settings, they might inadvertently get downsampled to screen resolution.

Inappropriate use of jpeg:



The left picture is a jpeg of 1138 bytes, the right one a png of 571 bytes. Jpeg is fine for photographs, but if your image contains large solid areas and sharp transitions, then lossless compression such as used by the png format is probably better.

Some of these problems can be spotted by zooming in on your figures in the Reader.

Page size and other properties

With the traditional LaTeX plus dvips plus Distiller route, you needed to tell all three programs about the desired page size. With pdftex, you only need to specify page dimensions once, in your TeX source. Use the pdftex primitives `\pdfpagewidth` and `\pdfpageheight`, or use the geometry package.

While you are at it, ensure also that the PDF version is no higher than it needs be, since your printshop may not have the latest versions of everything. A good version to aim for is version 1.3, which corresponds to Acrobat 4. This can be set either in `pdftex.cfg` or in your TeX source:

```
\pdfoptionpdfminorversion=3
```

Again, you can check either with the Reader, using either File/Document Properties/Summary or the Document Summary tab under the right-pointing arrow above the vertical scrollbar; see figure 2.

With the xpdf utilities, use `pdffinfo`:

```
> pdffinfo siepstyle.pdf
Title:          siepstyle
Creator:       TeX
Producer:      pdfTeX-1.11b
CreationDate:  20040601
ModDate:      20040601
Tagged:       no
Pages:        3
Encrypted:    no
Page size:    595.3 x 756 pts
File size:    148171 bytes
Optimized:    no
PDF version:  1.3
```

Page dimensions (pts) are in ‘big points’.

Combining documents

With a journal or a proceedings, it often isn’t practical to compile the entire document in a single TeX run. So you may end up with a separate pdf for each paper, which you have to combine into a single pdf somehow.

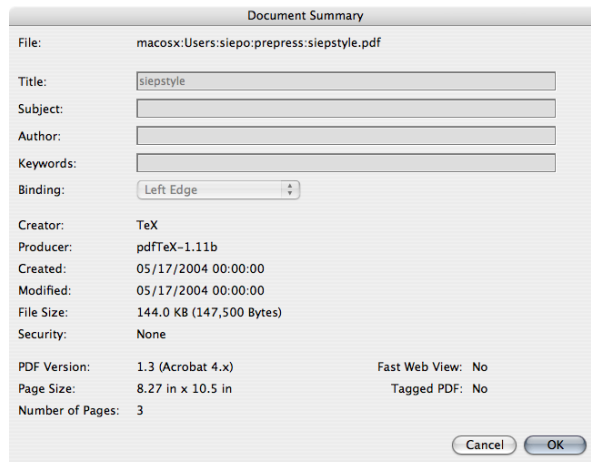


Figure 2. Adobe Reader: page dimensions and PDF version

With TeX. If you have separate pdfs of arbitrary origin then TeX can collate them for you: either use the LaTeX package `pdfpages` or use the ConTeXt utility `texexec` with the `--pdfarrange` switch. Including a file with pdfpages can be as simple as

```
\usepackage{pdfpages}
...
\includepdf [pages=--]{APaper}
\includepdf [pages=--]{AnotherPaper}
...
```

The `teTeX`/`fpTeX`/`TeX Live` distributions contain the necessary documentation for `pdfpages` and `texexec`.

With a Distiller driver file. Another option is to generate PostScript files and feed Distiller a driver file which loads them. Such a driver file may look as follows:

```
%!
/prun {
  /mysave save def      % save first
  dup = flush          % Shows name of PS file
  RunFile              % builtin Distiller proc
  clear cleardictstack % Cleans up
  mysave restore       % Restores save level
} def

(c:/temp/apaper.ps) prun
(c:/temp/anotherpaper.ps) prun
...
```

This is documented in the Acrobat documentation; see `RunDirEx.txt` and `RunFileEx.ps`. The location of these files vary per version and platform.

If you use this approach, it is best *not* to let dvips subset fonts. Then Distiller can create for each font a single subset for the entire volume, leading to a smaller pdf.

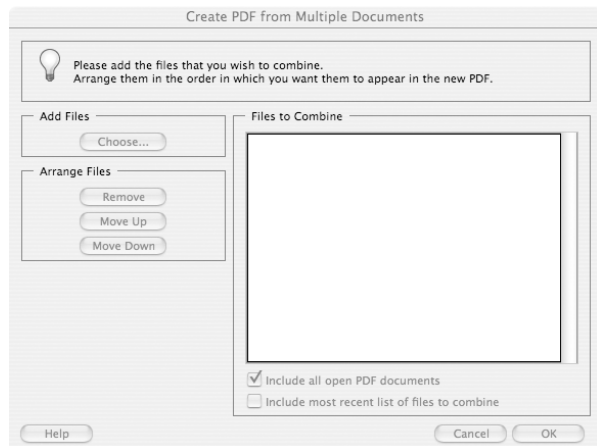


Figure 3. Combining pdfs interactively with Acrobat

With Acrobat. Finally, Acrobat lets you combine pdfs interactively, but since you probably end up repeating it quite a few times, the other options are almost certainly more convenient. See figure 3.

Color separation

If you want your document to be printed in color, then the printshop has to prepare one plate for each ink. For ‘full color’, these inks are usually cyan, magenta, yellow and black (CMYK). This style of color printing is called process color. The best way by far is to let the printshop handle this itself. After all, they should have the specialized software and the know-how.

However, T_EX users do have a few options:

Using macros. You can generate a page several times, each with different definitions for colors:

```
\def\doseparation#1{%
  \ifcase #1 % composite
    \def\sepcyan{cyan}%
    \def\sepblack{black}%
    \def\sepfigure{CKfigure}%
  \or % cyan
    \def\sepcyan{black}%
    \def\sepblack{white}%
    \def\sepfigure{Cfigure}%
    % cyan rendered as black; black omitted
  \or % black
    \def\sepcyan{white}%
    \def\sepblack{black}%
    \def\sepfigure{Kfigure}% cyan omitted
  \fi
  {\color{\sepcyan} Text in cyan\par}
  {\color{\sepblack} Text in black\par}
  \includegraphics{sepfigure}\newpage}

%\doseparation0
```

```
% for colored output; omitted for separations
\doseparation1
\doseparation2
```

Note that this requires pre-separated external figures.

ConT_EXt has macro-based color separation functionality built in; see www.pragma-ade.com/general/manuals/msplit.pdf.

Using dvips and colorsep.pro. The T_EX Live distribution contains a PostScript header file `texmf/dvips/colorsep/colorsep.pro` for separation of process colors. If you run `dvips` as follows:

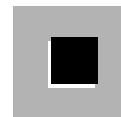
```
dvips -b 4 -h colorsep.pro filename
```

then `dvips` produces each page four times (`-b 4` switch), and each time the header file `colorsep.pro` re-defines colors appropriately for each of the four printing plates.

Using Acrobat 6 Professional. Acrobat 6 Professional also offers color separation via the Print menu. I encountered some glitches so I recommend to have a really good look at the resulting PostScript- or pdf file before submitting it to your printer.

Overprinting

When printing black over a colored background, color separation software usually sets the other plates to white. However, any misregistration on the press will lead to slivers of white, which might be quite distracting; see the picture below.



If the background is light enough, then you can ignore the effect, but in other cases it is better to do something about it. One solution is to use a modified black with other color components added:

```
\color[cmyp]{0,0.5,0,1}
```

Another solution is to tell PostScript or PDF to let the color continue underneath the black. This is called overprinting. For a L^AT_EX stylefile and example which *tries* to implement this for `dvips` and `pdftex`, look at <http://tex.aanhet.net/overprint/>. You can judge the effect in Acrobat Pro, if you check Advanced/ Separation Preview. Figure 4 shows this dialog in another context.

Spot colors

A popular use of color in a printed document is to print some elements such as headings or rules from

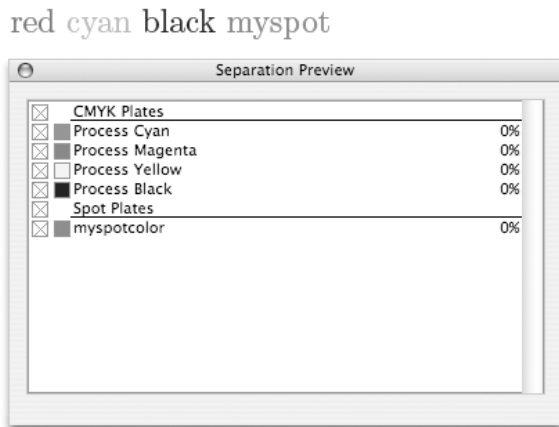


Figure 4. ConTeXt does spotcolors in addition to CMYK

a single premixed color. Printshops have books with color swatches to choose from. Pantone is the manufacturer and license holder of most of these swatch books. You can let one of the process colors, i.e. cyan, magenta or yellow, take the place of the spot color and tell the printshop which color you really want.

If you want spot color *in addition to* process color, then the above trick can't be used. However, ConTeXt offers real support for spot colors. You can do it as follows:

```
\definecolor[myspotcolor][c=.7,m=.2]
\definecolor[myspot][myspotcolor][p=1]
...
\color[myspot]{myspot}
```

Note the two-stage definition of `myspot`: if you want a separation plate for the spot color, you need to define `myspot` as a tint or fraction of a previously defined color. See also figure 4.

Color management

Rgb colors are represented by three values for the three components; process color by four values for the four process inks. These three or four values don't represent color itself but instructions for a device to apply certain colorants. The resulting color can and does depend on the device; we are all familiar with a wall of TV sets in an electronics store each displaying the same image with a different color cast. Matching screen colors with printed colors is a worse problem. We all have seen how screen images can become disappointingly dull when printed; many brilliant screen colors simply cannot be reproduced in print.

Because graphics professionals tend to care about color consistency, color management systems have been introduced, which try to guarantee color consistency from device to device. This means either specify-

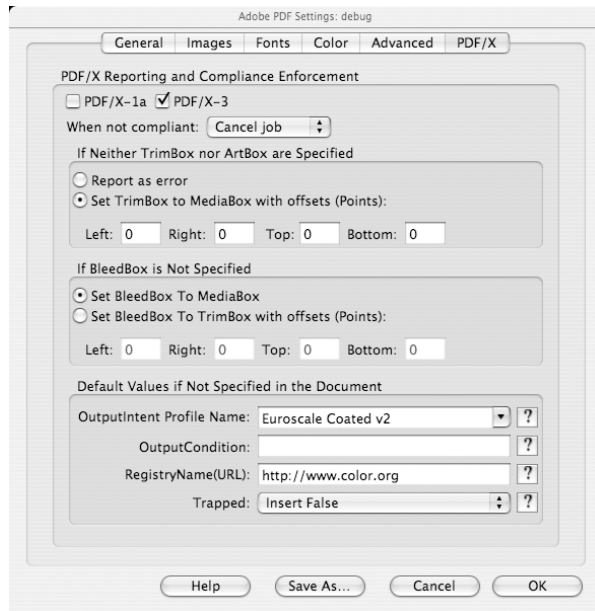


Figure 5. Distiller settings for PDF/X conformance. The Color tab (not shown) also contains relevant settings.

ing color in some device-independent way or to supply device profiles to go with the color elements in your document. This is one area where open source doesn't have much to offer.

PDF/X and Certified PDF

PDF/X is an ISO standard for pdfs in prepress. There are two flavors: PDF/X-1a which allows process color and spot color, and PDF/X-3 which also accepts color-managed RGB. Since it is an ISO standard, you have to pay money to get the specification. However, you can download documentation and Distiller settings for free from www.pdf-x.com.

If you can avoid RGB color altogether, then it is possible to generate PDF/X with pdftex. However, don't convert existing images just for the sake of PDF/X conformance if you don't have to; check with your printshop first.

Code similar to the following should ensure that your pdf won't fail PDF/X for silly reasons:

```
\pdfpagewidth=595.3bp
\pdfpageheight=841.7bp
\pdfpageattr{/TrimBox [ 0 0 595.3 841.7] }
\pdfoptionpdfminorversion=3

\edef\pdfdate{%
  \the\year
  \ifnum \month < 10 0\the\month \else \the\month \fi
  \ifnum \day < 10 0\the\day \else \the\day \fi}
\pdfinfo{%
```

```

/CreationDate (D:\pdfdate)
/ModDate (D:\pdfdate)
/Trapped (False)
/GTS_PDFXVersion (PDF/X-3)
/Title (\jobname)}

\pdfcatalog{
/OutputIntents [ <<
/Info (Euroscale Coated v2)
/Type /OutputIntent
/S /GTS_PDFX
/OutputConditionIdentifier (OFCOM_PD_P1_F60)
/RegistryName (http://www.color.org/)
>> ]}

```

Actually, pdftex 1.11b already includes a creation date automatically. Hopefully, newer versions will do the same for modification date so that you can dispense with the date rigmarole altogether.

Acrobat Distiller also has options for color management and PDF/X; see figure 5.

Another initiative, from Enfocus Software, is Certified PDF. This is not just a set of requirements, but requires your pdfs to be stamped as certified by dedicated commercial software. I found no reference to this type of certification in the Acrobat documentation. See www.certifiedpdf.net for more information.

Preflight

The term preflight has become in use for ensuring that your pdf is safe for production. I already mentioned a few simple checks that are available with the Reader and with the xpdf utilities.

Acrobat Professional has a lot of preflight options built-in, including checks on PDF/X compliance. They can be found under the Document menu. Just as with Distiller options, there are also named sets of preflight options; see figure 6.

Much of the Acrobat preflight code has been taken from Callas' PDF/X Inspector. There also used to be a free version of this tool, called PDF/X-3 Inspector.

Conclusion

The main points are to check what you can, and to discuss with your printshop in what form they want your document. Maybe they have a Distiller options file; even if you don't use Distiller, then it would still be useful to look at; these are plain ascii files.

If there is color then it is highly desirable that the printshop is willing to do the separations. The same is

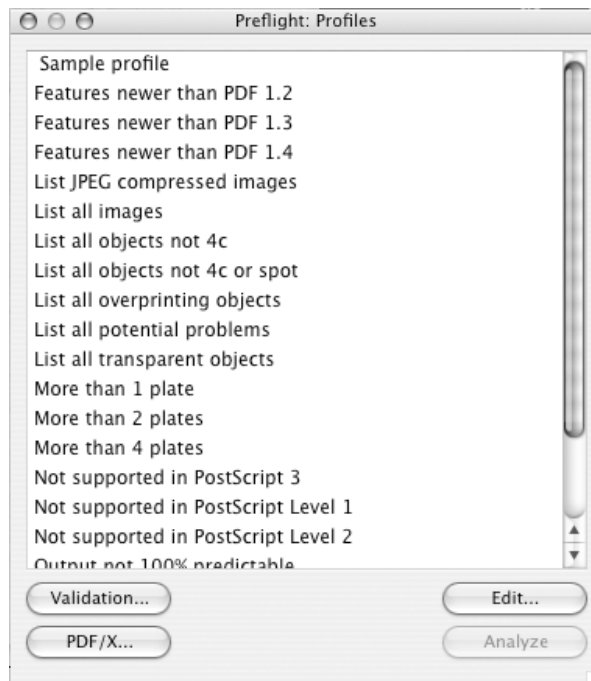


Figure 6. A long list of predefined sets of preflight profiles in Acrobat 6 Pro

true for page imposition.

Keep also in mind that there are lots of MS Office files which are being typeset somehow, so many printshops ought to be able to handle pdfs from outside the graphics industry.

All this being said, I do believe that Acrobat Pro is a worthwhile investment if you can afford it at all.

URLs

Adobe	www.adobe.com
Planet PDF	www.planetpdf.com
Xpdf	www.foolabs.com/xpdf/
DviPDFm project	project.ktug.or.kr/dvipdfmx/
Color separation in Context	www.pragma-ade.com/general/ manuals/msplit.pdf
overprint.sty	tex.aanhet.net/overprint/
PDF/X support	www.pdf-x.com
Certified PDF	www.certifiedpdf.net
Callas	www.callas.de/en/

Siep Kroonenberg
siepo@cybercomm.nl