



Weaving
PS into PDF

1 Introduction

This manual will describe the PSTOPDF conversion tool, written in RUBY on top of GHOSTSCRIPT. When going from POSTSCRIPT to PDF, you can use DISTILLER and a few other commercial applications, as well as the free GHOSTSCRIPT tool. Each tool has its advantages and in professional workflows they will be used intermixed.

The PSTOPDF converter evolved out of the converter built into T_EXUTIL, a file utility that comes with CON_TE_XT. That converter itself was inspired by the PDF to POSTSCRIPT pipeline posted by Sebastian Rahtz in the early days of PDF_TE_X.

Where T_EXUTIL goes beyond piping by applying some filters to the POSTSCRIPT code, PSTOPDF adds even more filtering and cleanup of POSTSCRIPT code, especially application specific directives that fall outside the scope of standards and converters.

The extensions were driven by the fact that in our projects we were confronted with rather messy POSTSCRIPT and cleanup was needed anyhow. Not only the content can be a problem, also filenames and suffixes can be messed up. This is why PSTOPDF is accompanied by a series of other image handling tools, described elsewhere.

2 The command line

Because this tool is part of the EXAMPLE toolkit that comes with CON_TE_XT, it's control may look strange but suits this framework.

```
--debug      shows a lot of extra info
--nopipe     avoids piping, handy for buggy platforms
--method     1-5, more on that later
--offset     an additional offset to cropped images
--prefix     tokens to prepend to the output filename
--suffix     tokens to append to the output filename
--inputpath  the path where input files are looked up
--outputpath the path where output files are placed
```

An example of usage is:

```
pstopdf --method=2 *.eps *.ai
```

3 Watched folders

You can set up a watched folder (aka hot folder) structure. This has the advantage that you only have to install one instance of PSTOPDF in a network.

`--watch` folder to watch for files to convert
`--delay` interval between watch scans

Say that we called PSTOPDF as follows:

```
pstopdf --watch=e:/pstopdf/watch --delay=2
```

The program will look in subfolders of the `watch` path for user subpaths, like:

```
.../pstopdf/watch/hagen
```

Results will be placed in:

```
.../pstopdf/watch/hagen/result
```

A converted file ends up in:

```
e:/pstopdf/watch/hagen/done
```

You can control the conversion by putting the files into the automatically created substructure:

```
.../pstopdf/watch/hagen/raw
.../pstopdf/watch/hagen/bound
.../pstopdf/watch/hagen/crop
```

There is an bonus subpath:

```
.../pstopdf/watch/hagen/bitmap
```

When bitmap images are placed on this path, they are converted to PDF using `IMAGEMAGICK`. The compression and quality chosen roughly depends on the size of the file.

4 Batch control

To do: Here we will discuss how you can set up conversion presets and or use `EXAMPLEX` to control `PSTOPDF`.

5 Configuration

There is not much configuration needed. First of all you need to install `RUBY`, `GHOSTSCRIPT`, and optionally `IMAGEMAGICK`. Then you must locate `PSTOPDF` in your `TEX` tree.

You can either call this script directly or call it by passing the filename to RUBY. This depends on your operating system and configuration.

```
.../tex/texmf-local/context/ruby
```

Details of the environment can be set up in the EXAMPLE configuration files, located in (for instance):

```
.../tex/texmf-local/context/exemplap/scripts
```

The main configuration file is called `example.exe`, which calls up a few others. Don't mess around with the file `basic.exe`, but stick to adapting `path.exe` or `local.exe` if needed.

6 Methods

Currently there three conversion methods for EPS images, and two for PDF images:

method	suffix	prefix	action
1	eps		leave boundingbox as it is
2	eps		crop image to internal bounding box
3	eps		crop image to analyzed bounding box
4	pdf	lowres-	downsample image to screen resolution
5	pdf	normal-	cleanup image to prepress defaults

When no output filename is specified, the prefix is appended to the filename.

7 The GUI

We've written an interface to this tool, called `pdtopdf.pdf`. You can find the interface in the path:

```
.../tex/texmf-local/context/exemplap/gui
```

The interface is more or less self documenting.