

ConTeXt

conversion to PDF

category: METAPOST

version: 2000.03.27

date: May 23, 2000

author: Hans Hagen

copyright: PRAGMA / Hans Hagen & Ton Otten

The file `mptopdf` provides a quick way to convert METAPOST files to PDF using a slightly stripped down plain \TeX , PDF \TeX , and a few Con \TeX t modules.

First generate a format, which in WEB2C looks like:

```
pdftex --ini mptopdf
```

Since this conversion only works with PDF \TeX or PDF- ϵ - \TeX , the session is aborted when another \TeX is used. When finished, the resulting `fmt` file should be moved to the right location.

The conversion itself is accomplished by:

```
pdftex &mptopdf \relax filename.number
```

The `\relax` is needed since we don't want to process the file directly. Instead we pick up the filename using `\everypar`. Since this file is still the first one we load, although delayed, the jobname is as we expect. So, at least in WEB2C, the result of the conversion comes available in the file `filename.pdf`. This conversion process is roughly compatible with:

```
texexec --pdf --fig=c --result=filename.pdf filename.number
```

This uses Con \TeX t, and is therefore slower. Therefore, we provide a small PERL script that does a faster job, using the minimal format. Given that a format is generated, one can say:

```
mptopdf somefile
mptopdf somefile.123
mptopdf mp*.*
```

The results are copied into files named `somefile-number`. This mechanism will also be available in a next release of \TeX UTIL.

The \TeX implementation is rather simple, since we use some generic Con \TeX t modules. Because we need a few register allocation macros, we preload plain \TeX . We don't load fonts yet.

```
1 \input syst-tex
```

We check for the usage of PDF \TeX , and quit if another \TeX is used.

```
2 \ifx\pdfoutput\undefined
  \message{Sorry, you should use pdf(e)TeX instead.}
  \expandafter \endinput
\fi
```

The conversion to PDF is carried out by macros, that are collected in the file:

```
3 \input supp-pdf
```

We use no output routine.

```
4 \output{}
```

Since we need to calculate and set the bounding box, we definitely don't want to indent paragraphs.

```
5 \parindent=0pt
```

We use `\everypar` to pick up the filename and process the METAPOST graphic.

6 `\everypar{\processMPfile}`

The main macro shows a few PDF_T_E_X primitives. The main work is done by the macro `\convertMPtoPDF` which is defined in `upp-pdf`. This macro interprets the `METAPOST` file. Close reading of this macro will probably learn a few (PDF) tricks. Apart from some path transformations, which are needed since PDF has a different vision on paths, the graphic is positioned in such a way that accuracy in PDF xforms is guaranteed.

```
7 \def\processMPfile#1 %
  {\pdfoutput=1
   \hsize100in
   \vsize\hsize
   \setbox0=\vbox{\convertMPtoPDF{#1}{1}{1}}%
   \ifdim\wd0<1in \message{[warning: width<1in]}\fi
   \ifdim\ht0<1in \message{[warning: height<1in]}\fi
   \pdfpageheight=\ht0
   \pdfpagewidth=\wd0
   \voffset=-1in
   \hoffset=\voffset
   \box0
   \bye}
```

Since acrobat has troubles with figures smaller than 1 inch, we issue a warning. When embedding graphics in documents, a size less than 1 inch does not harm.

The resulting PDF file is about as efficient as such a self contained file can be. However, if needed, this PDF file can be converted to EPS using for instance the `pdftops` program (in `WEB2C`) or `GHOSTSCRIPT`.

8 `\dump`