

latexMP

Jens-Uwe Morawski

(morawski@gmx.net)

April 6, 2005 – version 1.2

Contents

1	Intro	1
2	Getting started	1
3	Configuration	2
4	Special Features	4
5	latexMP and other packages	4
6	Examples	5
6.1	Customize the L ^A T _E X Preamble	5
6.2	Using Color Switches in L ^A T _E X Text	6
6.3	Using MetaFun for CMYK-Colors and Transparency	6
6.4	Customize the Graph package with latexMP	7

1 Intro

The METAPOST package latexMP implements a user-friendly interface to access L^AT_EX-based typesetting capabilities in METAPOST. The text to be typeset is given as string. This allows even dynamic text elements, for example counters, to be used in labels. Compared to other implementations it is much more flexible, since it can be used as direct replacement for btex..etex, and much faster, compared for example to the solution provided by tex.mp.

2 Getting started

The following METAPOST code shows an almost minimal latexMP example. In a loop five different labels are created that simply show the value of the counter used in the loop.

```
input latexmp ;
beginfig(1);
  for i=1 upto 5:
    label(texttext("$i=" & (decimal i) & "$"), (0,10*i)) ;
  endfor;
endfig;
end
```

The macro `texttext` is the interface to `latexMP`. It takes a string as the only parameter. Behind the scenes `latexMP` collects all strings given to `texttext` and writes them in a temporary file. The name of this file depends on the name of your `METAPOST` input file. It has the same base-name but with a prefix `ltx-` and the file extension `.tmp`; so if your input file is `figures.mp` then the temporary file will be `ltx-figures.tmp`.

In order to typeset the strings the temporary file has to be input again. This will be done automatically in a subsequent `METAPOST` run. This means, you need always two `METAPOST` runs in order to update changed and new labels.

3 Configuration

The \LaTeX preamble Typesetting labels with `latexMP` is only creating a \LaTeX document in a different way, but the document structure is almost the same. Therefore, `latexMP` allows to specify a preamble for the \LaTeX document used to typeset your labels. The macro `setupLaTeXMP` implements a key-value interface in order to configure different parts of the \LaTeX preamble.

A \LaTeX preamble consists of the declaration for the document class, `\usepackage` statements that load \LaTeX packages and other \LaTeX code. The document class can be specified using key `class`. If no document class has been specified the `article` class will be used. Options for the document class can be given using key `options`. Packages can be loaded with key `packages`.

```
setupLaTeXMP(class="report",
             options="12pt",
             packages="helvet[scaled=0.96],mathpazo");
```

As you can see, the values have to be strings. The above example results in the following \LaTeX preamble:

```
\documentclass[12pt]{report}
\usepackage[scaled=0.96]{helvet}
\usepackage{mathpazo}
```

There are some packages in \LaTeX that are frequently used especially in a non-english environment. Therefore, `latexMP` provides a direct interface to these packages. Additionally, one might pass verbatim \LaTeX code in the preamble. Extending above example:

```
setupLaTeXMP(class="report",options="12pt",
             fontencoding="T1",inputencoding="latin1",
             language="ngerman",
             preamble="\newcommand{\IT}[1]{\textit{#1}}",
             packages="helvet[scaled=0.96],mathpazo");
```

results in the following preamble:

```
\documentclass[12pt]{report}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage[ngerman]{babel}
\usepackage[scaled=0.96]{helvet}
\usepackage{mathpazo}
\newcommand{\IT}[1]{\textit{#1}}
```

In many situations the METAPOST figures are used in other documents and the style of the figures should match the document style. Thus, it makes sense to share the configuration of the document with the L^AT_EX typesetting routine in METAPOST. The key *preamblefile* allows to specify a L^AT_EX file that includes a valid L^AT_EX preamble. This preamble file will be loaded by latexMP instead of generating an own preamble. Consider a file `preamble.tex` containing the following code:

```
\documentclass[12pt]{report}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage[ngerman]{babel}
\usepackage[scaled=0.96]{helvet}
\usepackage{mathpazo}
```

In order to use this preamble with latexMP the METAPOST file can be configured as:

```
input latexmp ;
setupLaTeXMP(preamblefile="preamble");
beginfig(1);
  label(texttext("Ein kurzer Text."),origin) ;
endfig;
end
```

The document file then looks like:

```
\input{preamble}
\usepackage{color,graphicx}
\begin{document}
\includegraphics{figures.1}
\end{document}
```

As you can see, the packages *color* and *graphicx* are not part of the preamble file. In general, it is not a good idea to load L^AT_EX packages that use so called T_EX specials in a METAPOST-L^AT_EX preamble.

An alternative for the color package As already said, the L^AT_EX color package cannot be used in METAPOST since the specials it writes to the DVI-file get lost during the conversion from DVI to the METAPOST picture.

latexMP implements a very special way of tagging parts of text objects in order to support the interface of the color package. This requires some pre-processing of the METAPOST pictures before they are made available by `texttext`. Since this feature is not always required it has to be activated setting the parameter *multicolor* in `setupLaTeXMP` to `enable`. With this setup it is possible to use the L^AT_EX macros known from the *color* package, i. e. `\color` and `\textcolor`, in the strings given to `texttext`, see examples.

The color systems supported by this color implementation are *rgb*, *cmymk*, *gray*, *Hsv* and *named*. By default all colors, even a *cmymk*-color, are converted to *rgb*, since METAPOST only supports *rgb*.

rgb	r=0-1, g=0-1, b=0-1	<code>\textcolor[rgb]{1,0,0}{text}</code>
Hsv	H=0-360, s=0-1, v=0-1	<code>\textcolor[Hsv]{60,0.5,0.5}{text}</code>
gray	g=0-1	<code>\textcolor[gray]{0.6}{text}</code>
cmymk	c=0-1, m=0-1, y=0-1, k=0-1	<code>\textcolor[cmymk]{1,0,1,0}{text}</code>
named	uses defined METAPOST color	<code>\textcolor{colorname}{text}</code> or <code>\textcolor[named]{colorname}{text}</code>

4 Special Features

In this section special features of latexMP will be discussed. These features have to be used with care, since their activation can make your METAPOST file incompatible with the normal METAPOST behaviour.

Starting once, running twice As already known, latexMP writes the strings to be typeset in a temporary file. In the next METAPOST run these strings will be typeset automatically with L^AT_EX and are available as METAPOST pictures. Therefore two METAPOST runs are required to update changed or new labels. If the parameter *mode* is set to *rerun* in `setupLaTeXMP`, then latexMP automatically runs your input file twice. This is implemented by reading the input file at end again.

If you use latexMP stand-alone there should be no problem with this configuration. Nevertheless, other METAPOST packages can have problems, since in the second run the package is loaded again. Known packages that do not work in this mode are *MetaFun*, *MetaObj* and *graph*.

Other problems may occur if global variables are used. Linear equations solved at global level always get over specified, since in the second run these equations are solved already. latexMP provides some workarounds for some variables and configurations it knows, but it can not know what other packages or the user do. Thus, this feature allows to save run time, takes care of updating labels, but should used with care.

No infont anymore If one uses a string in the METAPOST label macro then this string will be converted in a picture using the `infont` operator. The typesetting capabilities of `infont` are very limited, thus it makes sense to replace it with `texttext`.

latexMP provides an alternative version of the internal METAPOST macro `thelabel`. Compared to the original macro it uses `texttext` instead of `infont` to typeset strings. The `thelabel` replacement provided by latexMP has to be activated using `texttextlabel=enable` in `setupLaTeXMP`. Now the string can be given to `label` directly without to use `texttext` anymore.

5 latexMP and other packages

MetaFun The normal METAPOST output is a simplified EPS format. Therefore it lacks some features, notably CMYK color support. The *MetaFun* package provides, besides many other useful macros, some extensions that are based on specials written in the METAPOST output. Therefore, it requires a post-processing of the METAPOST output with the

program *MPToPDF* in order to interpret those specials. The resulting PDF can then be embedded in a \LaTeX document.

The extensions interesting for latexMP are support for the CMYK color model and transparency. As said above, *MetaFun* is one of those packages that can not be used in *rerun* mode. Another requirement is, that *MetaFun* has to be loaded prior to latexMP .

If *MetaFun* has been loaded this way, then `cmlyk-colors` in the latexMP color support use automatically *MetaFun* specials to output real `cmlyk-colors` instead of `rgb`.

Additionally, it is possible to use a \LaTeX macro `\transparent` in your text to get transparency effects in your labels. The syntax is `\transparent[method]{value}{text}`. This makes the `text` transparent dependent on `value` using one of the methods supported by *MetaFun* and the PDF format. The default method is 1; for the other methods see the *MetaFun* manual.

6 Examples

6.1 Customize the \LaTeX Preamble

```
input latexmp ;

string OrdMacro, OoosMacro, otherDefs ;

OrdMacro :=
"\newcommand*\Ord}[1]{\ifcase #1\relax" &
  " #1\textsuperscript{th}\or" &
  " #1\textsuperscript{st}\or" &
  " #1\textsuperscript{nd}\or" &
  " #1\textsuperscript{rd}\else" &
  " #1\textsuperscript{th}\fi}" ;

OoosMacro :=
"\newcommand*\Ooos}[1]{\setcounter{Ooos}{1}0\whiledo{\value{Ooos}<#1}" &
  "{\ostepcounter{Ooos}}h}" ;

otherDefs := "\newcounter{Ooos}\renewcommand{\familydefault}{\sfdefault}" ;

setupLaTeXMP(
  inputencoding="latin1",
  fontencoding="T1",
  packages="helvet,calc,ifthen,mflogo",
  preamble=(OrdMacro & OoosMacro & otherDefs) ,
  textxlabel=enable );

beginfig(1);
  for i=1 upto 5:
    label.urt( "\Ord{" & (decimal i) & "} {\MP} label", (0,-10*i) ) ;
  endfor;
endfig;
```



```

        "\transparent{0.5}{with a red transparent overlay}", origin-(0,2mm)) ;
endfig;
end

```

A short green text in RGB.
with a red transparent overlay
A short green text in CMYK.

6.4 Customize the Graph package with latexMP

```

input latexmp ;
input graph ;

% create data file
for D="10 2e-2","10 2e-2","20 3e4" ,"30 8e4","100 7e4","130 2e5",
    "200 4e5","500 8e5","600 1e7","1000 2e7","2000 8e7":
    write D to "lmp-data.dat";
endfor;
write EOF to "lmp-data.dat" ;

setupLaTeXMP(
    mode=normal,% 'graph' package does not work in rerun mode
    packages="mathpazo") ;

init_numbers(texttext("$-$"),texttext("$1$"),texttext("${\times}10$"),
    texttext("${}^-$"),texttext("${}^2$")) ;
Fe_base:=texttext("$10$") ;

beginfig(1);
    draw begingraph(6cm,4cm)
        setcoords(linear,log);
        gdraw "lmp-data.dat" ;
    endgraph ;
endfig;
end

```

