MetaPost in ConTEXt

1 Summary

This talk:

I Gives a little bit of history of using MetaPost.

II Summarizes how we currently deal with mplib.

III Provides some examples of extensions.

2 The status quo

A short history of the integration:

I Soon after MetaPost showed up ConTEXt started providing integrated support.

II Conversion to pdf (backend code) is built into the ConTEXt core.

III Delayed or instant processing is part of MkII but MkIV only has instant processing.

IV Some data is passed between ConTEXt and MetaPost.

V Text processing is related to the main run which brings consistency.

VI Macros are wrapped in a package called MetaFun.

VII Users have adopted MetaPost so there is support on the list.

3 Extending MetaPost

Rather soon MetaFun provided some extensions, like:

I Support for the cmyk color space.

II Support for spot and multitone colors (needs backend resource management).

III Transparency (relates to colors).

IV Fills that can have linear and circular shading.

V Inclusion of external images.

VI Embedding and manipulation of outline text.

VII Access to the ConTEXt positioning mechanism.

The implementation uses specials, colors and dummy paths.

4 Moving to MPLIB

The mplib project made it possible to move forward:

I Development is now focussed on LuaTEX and MkIV.

II There we use the embedded mplib exclusively.

III All processing is now runtime.

IV Format files are not used any more.

Concerning the extensions:

I The first version of the conversion code used specials.

II The next version used pre- and postscripts for some extensions.

III The current version uses only `withprescript "something" withpostscript "anything"` directives.

5 Entering the next stage

Recently the extension code has been overhauled:

I We often have rather resource demanding MetaPost graphics, especially our new-year cards.

II Even more demanding are some of Mojca's gnuplot graphics.

III When we ran into memory problems for gnuplot output the decision was made to move away from specials.

IV In the process we found out that the memory problems were actually a bug in mplib (fixed already).

V But the (delayed) move was a good one as mixing solutions is somewhat messy.

VI And then of course Mojca became more demanding so now we have independent transparencies and bitmaps as well.

6 Towards an extension framework

I Already in MkII it was possible to add extensions.

II But accumulating them was not that trivial.

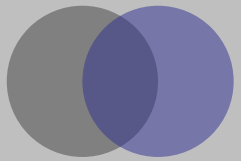
III The current approach is a bit more robust (no overload of colors).

IV Some extensions currently are multipass (like outlines).

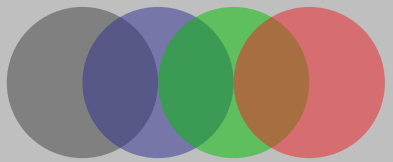
V Eventually the extensions will also support svg (as part of the export code).

7 Transparency

```
fill fullcircle scaled 2cm
   withcolor .5white ;
fill fullcircle scaled 2cm shifted (1cm,0)
   withcolor transparent(1,0.5,(1,1,0,0)) ;
```



```
fill fullcircle scaled 2cm shifted (2cm,0)
    withcolor 0.75*transparent(1,0.5,green) ;
fill fullcircle scaled 2cm shifted (3cm,0)
    withcolor (0,1,1,0) withtransparency (1,0.5) ;
```



8 Spot colors

```
\definecolor [blue] [c=1,m=.38,y=0,k=.64] % pms 2965 uncoated m
\definespotcolor [blue-100] [blue] [p=1]

fill fullcircle scaled 2cm
    withcolor \MPcolor{blue-100} ;
fill fullcircle scaled 2cm shifted (3cm,0)
    withcolor .25*\MPcolor{blue-100} ;
fill fullcircle scaled 2cm shifted (6cm,0)
    withcolor .50*\MPcolor{blue-100} ;
```

9 Multitone colors

```
\definecolor [yellow] [c=0,m=.28,y=1,k=.06] % pms 124 uncoated m
\definespotcolor [yellow-100] [yellow] [p=1]
\definemultitonecolor [mix] [blue=.5,yellow=.75] % [c=.1,m=.2,y=.3,k=.4]

fill fullcircle scaled 2cm
    withcolor \MPcolor{somecolor} ;
fill fullcircle scaled 2cm shifted (3cm,0)
    withcolor transparent(1,0.5,\MPcolor{blue-100}) ;
fill fullcircle scaled 2cm shifted (4cm,0)
    withcolor transparent(1,0.5,\MPcolor{yellow-100}) ;
fill fullcircle scaled 2cm shifted (5cm,0)
    withcolor transparent(1,0.5,\MPcolor{mix}) ;
```

10 Typeset text

```
draw textext("Hello, does this work?") ;
draw textext("\bf Hello, does this work?") shifted (0,-8mm) ;
```

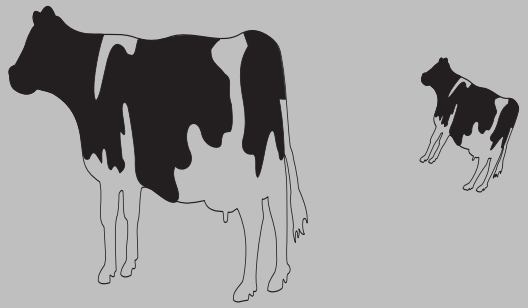
Hello, does this work?

Hello, does this work?

```
draw textext("\bfd Hello, {\blue does} this work?")
    rotated  10 withcolor white withtransparency (1,0.5);
draw textext("\bfd Hello, {\green does} this work?")
    rotated -10 withcolor white withtransparency (1,0.5);
```

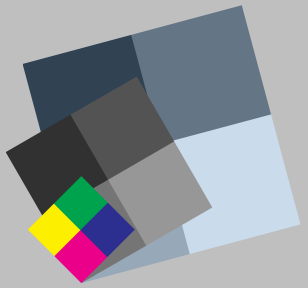
11 External figures

```
draw externalfigure "cow.pdf" xsized 4cm ;
draw externalfigure "cow.pdf" rotated -25 xsized 2cm shifted (5cm,2cm) ;
```



Bitmaps 12

```
draw
    bitmapimage(2,2,"334455 667788 99aabb ccddee")
    scaled 3cm rotated 15 ;
draw
    bitmapimage(2,2,"33 55 77 99")
    scaled 2cm rotated 30 ;
draw
    bitmapimage(2,2,"0000ff00 ff00ff00 00ff0000 ffff0000")
    scaled 1cm rotated 45 ;
```



13 Image Masks

```
draw externalfigure "mask-001.png" ysized 2cm ;
```

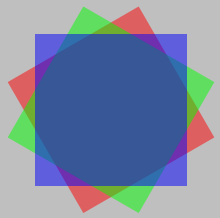
```
draw externalfigure "mask-002.png" ysized 2cm ;
```

```
draw externalfigure "mask-001.png" ysized 2cm withmask "mask-002.png" ;
```

14 Viewerlayers

```
\defineviewerlayer[rotation:30]
\defineviewerlayer[rotation:60]
\defineviewerlayer[rotation:90]

fill fullsquare scaled 2cm rotated 30 withcolor red   withtransparency(1,.5)
    onlayer "rotation:30" ;
fill fullsquare scaled 2cm rotated 60 withcolor green withtransparency(1,.5)
    onlayer "rotation:60" ;
fill fullsquare scaled 2cm rotated 90 withcolor blue  withtransparency(1,.5)
    onlayer "rotation:90" ;
```



Toggle 30 Toggle 60 Toggle 90

Shades 15

```
path    p ; p := fullcircle scaled 2cm ;
numeric s ; s := define_circular_shade(origin,origin,0,2cm,.5white,green) ;
fill p withshade s ;
```

```
path    p ; p := fullsquare scaled 2cm ;
numeric s ; s := define_linear_shade(center p,urcorner p,.5white,red) ;
fill p withshade s ;
```