Metapost Developments

Abstract
The new release of metapost includes some new features as well as a number of bugfixes. The new functionality includes: the possibility to use a template for the naming of output files; support for cmyk and greyscale color models; per-object Postscript specials; the option to generate Encapsulated Postscript files adhering to Adobe's Document Structuring Conventions; the ability to embed re-encoded and/or subsetted fonts; and support for the GNU implementation of troff (groff).

Introduction

Version 0.901 of Metapost was released at BachoTEX 2005. It was mostly a bugfix release, that featured and updated manual and the new `mpversion` primitive on top of a set of bugfixes.

At that time, a new version was promised for the autumn. In hindsight, that was overly optimistic. It is now already the summer of 2006, and the feature set for version 1.0 is now finally frozen. It will be released in time for TEXLive 2006 and (hopefully MikTEX 2.5), so to an average user not much time will be lost by the delay.

Bugfixes

Stability issues

In previous versions of Metapost, the size of the memory array was not stored in mem file. But in Web2c–base systems, the memory sizes are dynamic and the size that should be used by the executable can change depending on the command–line invocation. This discrepancy resulted in a number of painful and unexpected bugs.

- Disappearing specials from the output
- Incorrect error messages
- Unexplained crashes

This problem will be tackled by storing the required minimum memory sizes in the memory dump file. If an unsolvable mismatch occurs, an error message will be issued.

turningnumber

The current (0.9) Metapost executable has a very simple algorithm to calculate the `turningnumber` operation. It simply connects the path's points using straight segments, adds up all the angles between those segments, and then divides the result by 360. This only works well if the path segments are well-behaved i.e. they do not self-intersect.

This is already an improvement over the old code in the sense that when it is wrong, it is predictably wrong. But it was a temporary measure, and the next version contains completely new code that calculates true curvature for the path segments.

The new algorithm is based on a mailing list discussion between members of the group. It will be slower, but (finally) 100% accurate.

New features

File-name templates

The first of the new feature is support for output filename templates. These templates use `printf`-style escape sequences and are re-evaluated before each shipout. Numeric fields can be left-padded to a user-supplied width by prepended zeroes.
The new primitive command is `filenametemplate`, and it is a string-valued command. The syntax is as simple as:

```
filenametemplate "%j-%3c.eps";
beginfig(1);
  draw p;
endfig;
```

If the file is saved as `test.mp`, then this will create the output file `test-001.eps` instead of `test.1` of previous versions.

A small set of escape sequences are possible, see table 1 for details.

To ensure compatibility with older files, the default value of `filenametemplate` is `%j.%c`. If you assign an empty string, it will revert to that default.

CMYK color model

Support will be added for the industry-standard CMYK color model. In the simples form this looks like:

%%	A percent sign
%j	The current jobname
%⟨0-9⟩c	The charcode value
%⟨0-9⟩y	The current year
%⟨0-9⟩m	The numeric month
%⟨0-9⟩d	The day of the month
%⟨0-9⟩H	The hour
%⟨0-9⟩M	The minute

Table 1. Allowed escape sequences for `filenametemplate`

```
beginfig(1);
   draw fullcircle
      withcmykcolor (1,0,0,0);
endfig;
```

To make more flexible use possible, a new type of expression is introduced. A `cmykcolor` is a quartet of numerics that behaves just like the already existing type `color`.

```
beginfig(1);
   cmykcolor cyan;
   cyan := (1,0,0,0);
   draw fullcircle withcmykcolor cyan;
endfig;
```

The new `cyanpart`, `magentapart`, `yellowpart` and `blackpart` allow access to various bits of a `cmykcolor` or the CMYK component of an image object.

Greyscale color model

There are only two new primitives for greyscale support: `withgreyscale` and `greypart`. That is because greyscale values are simple numerics.

```
beginfig(1);
   faded := 0.5;
   draw fullcircle withgreyscale faded;
endfig;
```

An image object cannot have more then one color model, the last `withcolor`, `withcmykcolor` or `withgreyscale` specification sets the color model for any particular object.

RGB color model

Two new aliases for the already existing RGB color model will be added to plain.mp. You are requested to use these new keywords `rgbcolor` and `withrgbcolor` when referring to the old color model.

Object specials

The new Metapost will support two specials that can be attached to drawing objects. They are output on their own lines, immediately before and after the object they are attached to.

The new drawing options are `withprescript` and `withpostscript`, their arguments simple strings that are output as-is. It is up to the macro writer to make sure that the generated Postscript code is correct.

```
beginfig(1);
  draw fullcircle
     withprescript "gsave"
       withpostscript "grestore";
endfig;
```

Standalone EPS

If `prologues` is set to the value 2, Metapost will generate a proper Encapsulated Postscript level 2 image that does not depend on dvips postprocessing. In this output mode, fonts not be downloaded, but their definition will be handled correctly (see the next paragraph).

Thanks to a detailed set of comments by Michail Vidiassov, this output mode will adhere to Adobe's Document Structuring Conventions. A private Postscript dictionary will be created to reduce the output size for large images.

Font re-encoding

If `prologues` is set larger then 1, any used fonts are automatically re-encoded. Their encoding vectors will be included in the output if that needed.

This code is based on the font library used by dvips and pdfTEX. Following in the footsteps of pdfTEX, there are two new associated primitives: `fontmapfile` and `fontmapline`. The string-value argument has the same optional flag that is used by pdfTEX:

	replace the current font list completely
+	extend the font list, but ignore duplicates
=	extend the font list, replacing duplicates
−	remove all matching fonts from the font list

```
prologues := 2;
fontmapfile "+ec-public-lm.map";
beginfig(1);
  draw "Helló, világ" infont "ec-lmr10";
endfig;
```

Font inclusion

Font inclusion is triggered by `prologues` being equal to 3. Whether or not actual inclusion / subsetting takes place is controlled by the map files. These can be specified using the syntax explained in the previous paragraph.

GNU groff support

Version 1.0 of Metapost will have native support for GNU groff, thanks to a set of patches by Werner Lemberg and Michail Vidiassov.

Future plans

The next release after this one is likely to contain the following:

- A option to build metapost as embeddable library instead of an executable.
- 64-bit internal calculations instead of the current 32 bits.
- Alternative output formats for easier parsing by script backends
- The possibility to store drawing objects
- 12-part transform expressions to make it easier for macro packages to implement three-dimensional points.

Where to find Metapost

WWW Homepage and portal:
http://www.tug.org/metapost

User mailing list:
http://www.tug.org/mailman/listinfo/metapost

Development & sources:
https://foundry.supelec.fr/projects/metapost

Taco Hoekwater