

Compiling MetaPost figures under ConT_EXt

Abstract

To teach yourself MetaPost, the book “Learning MetaPost by Doing” by André Heck is probably unsurpassed. However, the examples therein are processed on Unix using L^AT_EX. ConT_EXt users have a bit of detective work to do before they can have successful compilations. If you are new to ConT_EXt, the lines below may help save your a few hours of experimenting.

These instructions were extracted from the MetaFun manual by Hans Hagen (from chapters 1, 2 and 3), and from a small macro that he once gave me that makes it possible to use the graph package by John Hobby.

Running MetaPost

1. Prepare a text file with the same example as Figure 1 in Heck:

```
beginfig(1);
draw (0,0)--(10,0)--(10,10)--(0,10)--(0,0);
endfig;

end.
```

Save it as `example.mp`

2. Compile this file under ConT_EXt with the following command:

```
texexec --mptex example.mp
```

which produces some temporary files plus the encapsulated Postscript file `example.1`

3. You may remove the temporary files with the command:

```
texutil --purge
```

but this is not necessary since they will be overwritten with the next compile run. Note that sometimes this command will not remove all temporary files, because some of those files are used by ConT_EXt to speed up a next compilation run.

4. The `example.1` file may be viewed with Ghostscript. Another way of viewing the results is to prepare a small file as follows:

```
%output=pdfTex
\starttext
\externalfigure[example.1]
\stoptext
```

and save it as `sample.tex`. Compile this with the command:

```
texexec sample
```

which produces (among other files) the output file `sample.pdf` which can be viewed by Acrobat. At the same time this shows how you can use MetaPost output in a T_EX document. Simply put it somewhere in your text file with the `\externalfigure[]` instruction.

5. If they bother you, you can again remove the temporary files by issuing the `texutil -purge` command.

MetaPost embedded in a ConT_EXt document

By embedding the MetaPost figure in a ConT_EXt document and compiling it, the whole process up to and including the creation of the resultant PDF file is achieved in a single processing step.

1. Type the following T_EX file:

```
%output=pdftex          % produces PDF, not DVI output
\noheaderandfooterlines % removes the page number
\starttext
\startuseMPgraphic{one}
  draw (0,0)--(10,0)--(10,10)--(0,10)--(0,0);
\stopuseMPgraphic
\useMPgraphic{one}
\stoptext
```

and save it as `emtest.tex`. Note that the file extension is now `tex` and not `mp`. Note also that the `beginfig()` and `endfig` are replaced by the `\startuseMPgraphic{}` and `\stopuseMPgraphic`. Finally note that the number of the figure has been replaced by a name, which is usually more easily remembered.

2. Compile it with the command:

```
texexec emtest
```

which produces the result file `emtest.pdf` which can be viewed by Acrobat.

Using Hobby's graph package

For plotting scientific data the graph package offers several facilities, such as plotting data from a file and automatic scaling. With proper labeling good looking graphs can be programmed.

1. Type the following program to produce a figure similar to the one part of the Figure 1 from the recent paper by Maarten Sneep in MAPS 31, page 12:

```
initialize_numbers;
input graph;
Autoform:="@g";

beginfig(0);
draw begingraph(9cm,4cm);
  gdraw"Apples.dat";
  autogrid(itick.bot,itick.lft);
  glabel(btex \ss\tfa Apples etex,1995,1360);
endgraph;
endfig;
end.
```

and save it as `hob.mp`. `initialize_numbers` is probably ConTeXt. I don't know what the `Autoform` is for. The `graph` package must be separately loaded. The `glabel` between the `btex .. etex` pair allows regular ConTeXt commands such as `\ss` for sans serif or font size changing ones such as `\tfa`. Note that only the label is now in sans serif style. The numbers along the axes are still in the regular roman style. The next section shows how to change that too. `Apples.dat` is a column oriented ASCII text file with numbers. The first column is the x-axis, the second column is treated as the y-axis data. An empty line in the file stops the data input to the plot.

2. Compile this file with the command:

```
texexec --mptex hob.mp
```

which produces the result file `hob.0` again as an encapsulated Postscript file. It can be viewed and included in a document as shown before.

The graph package in embedded form

This section describes a MetaPost program embedded in a ConTeXt file.

1. Type the following TeX file:

```
%output=pdfTeX language=en % causes PDF result
\noheaderandfooterlines % removes page number
\setupbodyfont[12pt]
\setupcolors[state=start] % no colors if absent

% Somehow needed with the \type{graph} package: -----
\forceMPTEXgraphictrue
\appendtoks
  initialize_numbers;
  input graph;
  Autoform:="@g";
\to \MPinitializations

% Gives sansserif numbers text and scale values: -----
\startMPinclusions
def do_initialize_numbers =
  if not numbers_initialized :
    init_numbers ( texttext.raw("$\hbox{\ss -}$"),
                  texttext.raw("$\hbox{\ss 1}$"),
                  texttext.raw("${\times}\hbox{\ss 10}$"),
                  texttext.raw("${}^{\hbox{\ss -}}$"),
                  texttext.raw("${}^{\hbox{\ss 2}}$") ) ;
    numbers_initialized := true ;
  fi ;
enddef ;
\stopMPinclusions

\startMPenvironment
  \switchtobodyfont[14pt,ss]
\stopMPenvironment

\startuseMPgraphic{two}
  path HLineone; HLineone:=(0,1)--(1200,1);
  path HLineten; HLineten:=(0,10)--(1200,10);
  draw begingraph(6cm,6cm);
    setcoords(linear,log);
    setrange(0,.1,1200,100);
```

```

glabel.lft(texttext.raw("$\hbox{\ss\tfb 0.1}$"),0,0.1);
glabel.lft(btex \ss\tfb 1 etex,0,1);
glabel.lft(btex \ss\tfb 10 etex,0,10);
glabel.lft(btex \rm\tfb100 etex,0,100);
glabel.lft(btex \rm 2.24 etex,1200,2.24);
gdraw HLineone withcolor .65white;
gdraw HLineten withcolor .65white;
gdraw "one.g" withcolor .65blue;
autogrid(otick.bot,);
endgraph;
\stopuseMPgraphic
\useMPgraphic{two}
\stoptext

```

and save this file as `hobby.tex`.

The paragraphs with `\appendtoks` and `\startMPinclusions` can be placed in a separate file named `m-graph.tex` and input to the program with the instructions:

```
\usemodule[graph]
```

ConT_EXt automatically adds the `m-` and loads the proper file.

2. Compile this file with the command:

```
texexec hobby
```

Conclusion

ConT_EXt has many more commands and instructions to work with MetaPost files. In addition Hans Hagen has programmed nice and useful extensions to MetaPost that can be found in his MetaFun manual and can be loaded with: `input m-tool`. To compile the examples in Heck's "Learning MetaPost by doing" I believe the above suffices. Programming in MetaPost is often a bit tedious but publication quality graphs can be obtained with patience, that are scalable, colorful, and of very great precision. Have fun.

Karel H. Wesseling