

Document Engineering with T_EX and METAPOST

Example Code

Ed Cashin

OIT
Feb. 16, 2000

Contents

- 1 background/test-static.mp 2
- 2 background/makefile 3
- 3 background/test-static.tex 4
- 4 test.pl 5
- 5 test.tex 9

Copyright 2000, Ed Cashin. All rights reserved.

All product names are trademarks of their producers.

T_EX sources for this document are available by request under terms of the GNU General Public License. Email the author for details at:

`ecashin@coe.uga.edu`

background/test-static.mp

```
1 % background.mp
2 %
3 % MetaPost code for static figure (of form's background)

4 % the stuff from "verbatim" to "etex" is passed to TeX for the
5 % things in this MetaPost figure that are processed by TeX (usually
6 % labels).

7 verbatim
8 \parindent=0in           % to avoid indentation
9 \font\twentyssbx=cmsbx10 at 20 true pt
10 etex

11 % drawrect is a macro

12 vardef drawrect(expr llft, urt) =
13   draw llft--(xpart llft, ypart urt)--urt--(xpart urt, ypart llft)--cycle;
14 enddef;

15 beginfig(1);
16   u = 1cm;                % u is a variable for units
17   drawrect((2u,2u),(12u,12u)); % big rectangle
18   drawrect((14u,2u),(16u,12u)); % smaller rectangle

19   label.rt(btex \twentyssbx STATIC etex, (2u, 13u));

20   % do some horizontal lines with a for loop

21   for i = 3 upto 11:
22     draw (2u, i * u)--(12u, i * u);
23   endfor;

24   % draw invisible dots to make sure the figure is big enough

25   pickup pencircle scaled .01; % microscopic pen
26   draw (origin);
27   draw (20u, 16u);
28 endfig;

29 end
```

background/makefile

2

background/makefile

```
1 DOC      = test-static
2 FIGS     = $(DOC).1

3 $(DOC).pdf : $(DOC).tex $(FIGS)
4             pdftex $(DOC).tex < /dev/null
5 $(DOC).ps : $(DOC).dvi
6             dvips -o $(DOC).ps $(DOC)
7 $(DOC).dvi : $(DOC).tex
8             tex $(DOC) < /dev/null

9 # (you can generate these rules like this)
10 # perl -e 'for(1..40){print "%.$_\t: %.mp\n\tmpost \${<\n";}'

11 %.1      : %.mp
12          mpost $< < /dev/null
```

background/test-static.tex

```
1 % this tex file is for producing testing documents that have the
2 % background on them

3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 \font\fourteenss=cmss10 at 14 true pt
5 \font\ninss=cmss10 at 9 true pt
6 \font\eightss=cmss10 at 8 true pt
7 \font\sevenss=cmss10 at 7 true pt
8 \font\sixss=cmss10 at 6 true pt
9 \font\fivess=cmss10 at 5 true pt
10 \font\fourss=cmss10 at 4 true pt
11 \def\bigspace{\quad\kern -.2em}
12 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

13 \def\thisproject{test-static}

14 \newdimen\tempdim
15 \tempdim=11in
16 \advance\tempdim by -12mm
17 \hsize \tempdim
18 \tempdim=8.5in\advance\tempdim by -5mm
19 \vsize \tempdim
20 \hoffset -.75in
21 \voffset -.85in

22 % here we test to see whether we're doing
23 % postscript or pdf:
24 \ifx\pdfoutput\undefined
25 \input epsf
26 \special{papersize=11in,8.5in}
27 \else
28 \input supp-pdf % mp support for pdftex
29 \pdfpageheight=8.5in % landscape letter pagesize
30 \pdfpagewidth=11in
31 \fi

32 \nopagenumbers
33 \parindent=0in

34 \ifx\pdfoutput\undefined
35 \epsfbox{\thisproject.1}
36 \else
37 \convertMPtoPDF {\thisproject.1} {1}{1}
38 \fi

39 \bye
```

test.pl

4

test.pl

```
1  #! /usr/bin/perl -w
2  # test.pl
3  #          typeset args onto PDF form

4  use strict;
5  use vars qw($TMP_DIR $ORIG_DIR $PROGBASE);

6  $TMP_DIR      = "/var/tmp";
7  $ORIG_DIR     = "/home/ecashin/src/tex/OITdocEngClass/example";
8  $PROGBASE     = "test";

9  &process(@ARGV);

10 sub process {
11     my @fields      = @_;
12     my $dir         = "$TMP_DIR/$PROGBASE-$$";

13     &xsystem("mkdir $dir", "mkdir call");
14     my $mp_source   = &make_mp_source($dir, \@fields);
15     &run_mpost($dir, $mp_source);
16     &symlinks_to_static($dir);
17     &run_tex($dir);
18     print &slirp_file("$dir/$PROGBASE.pdf");
19     &cleanup($dir);
20 }

21 sub make_mp_source {
22     die "Error: missing parameter" if @_ < 2;
23     my ($dir, $fieldsR) = @_;
24     my $basename       = "$PROGBASE-dyn.mp";
25     my $filename       = "$dir/$basename";
26     open MP, "> $filename"
27         or die "Error: could not open $filename for writing";

28     #-----backslashes must be escaped when using double quotes
29     my $MP_code        = <<"EAT_DAILY";
30     % $PROGBASE.mp
31     % MP code generated by $0

32     verbatimtex
33     \parindent=0in          % to avoid indentation
34     \font\twentyssbx=cmssbx10 at 20 true pt
35     \font\fourteenrm=cmr12 at 14 true pt
36     etex

37     beginfig(1);
```

test.pl

```
38   u = 1cm;
39   EAT_DAILY

40   my ($x, $y)                = (2, 2.5);
41   foreach my $field (@$fieldsR) {
42       #-----usually you'd have to use a lookup table for finding
43       #         the x and y for the field's position, but we use a
44       #         simple loop here.

45       $field                  = &tex_escaped($field);
46       $MP_code                 .= " label.rt(btex \\fourteenrm $field etex, "
47       . "({x}u, {y}u));\n";
48       $y                       += 1;
49   }
50   $MP_code                    .= <<"WORK_DAILY";

51   % use invisible dots at the edges to place the figure
52   pickup pencircle scaled .01; % microscopic pen
53   draw (origin);
54   draw (20u, 16u);
55   endfig;

56   end

57   WORK_DAILY

58   print MP $MP_code;
59   close MP;

60   return $basename;
61 }

62 sub run_mpost {
63     die "Error: missing parameter" if @_ < 2;
64     my ($dir, $mp_source)         = @_;

65     my $command                   = "cd $dir && mpost $mp_source < /dev/null";

66     &xsystem($command, "mpost call");
67 }

68 sub symlinks_to_static {
69     die "Error: missing parameter" if @_ < 1;
70     my ($dir)                     = @_;
71     my $command                   = "cd $dir "
72     . " && ln -s $ORIG_DIR/$PROGBASE-static.1 $PROGBASE-static.1 "
73     . " && ln -s $ORIG_DIR/$PROGBASE.tex $PROGBASE.tex ";

74     &xsystem($command, "symlinking call");
75 }
```

test.pl

```
76 sub run_tex {
77     die "Error: missing parameter" if @_ < 1;
78     my ($dir) = @_;

79     my $command = "cd $dir && pdftex $PROGBASE.tex < /dev/null";

80     &xsystem($command, "tex call");
81 }

82 sub slurp_file {
83     die "Error: missing parameter" if @_ < 1;
84     my $filename = shift;
85     open SLIRPFILE, $filename
86     or die "Error: could not open $filename for reading";
87     local $/ = undef;
88     my $file_contents = <SLIRPFILE>;
89     close SLIRPFILE;

90     return $file_contents;
91 }

92 sub cleanup {
93     die "Error: missing parameter" if @_ < 1;
94     my ($dir) = @_;
95     my $command = "rm -rf $dir";
96     &xsystem($command, "removing $dir call");
97 }

98 sub xsystem {
99     #-----wrapped system call
100    die "Error: missing parameter" if @_ < 2;
101    my ($command, $pidgeon) = @_;
102    if (my $error = system $command) {
103        $error /= 256;
104        die "Error: $pidgeon exited with abnormal value: $error";
105    }
106 }

107 sub tex_escaped {
108     # NOTE:
109     # this subr shows up a lot.
110     # this version is current as of Wed Feb 16 11:41:29 EST 2000

111     die "Error: missing parameter" if @_ < 1;
112     my $text = shift;
113     return "" if (! defined $text) or $text eq "";

114     study $text;
115     #-----handle quotes
116     while ($text =~ s/\\"'/ && $text =~ s/\\"'/) { }
```


test.pl

```
117 #-----escape special chars
118 #-----first are extra-special characters: do \, {, and } first
119 #-----so that our own TeX code doesn't get messed up

120 # \ is a special case, since using
121 # $\backslash$ would conflict with the
122 # rule for escaping '$'
123 $text      =~ s/\\/dollarbackslashadolla/g; # \
124 $text      =~ s/\$/\\$/g;                  # $
125 $text      =~ s/{/\$\lbrace$/g;           # {
126 $text      =~ s/}/\$\rbrace$/g;           # }

127 $text      =~ s/&/\\&/g;                   # &
128 $text      =~ s/^\char'136\relax /g; # ^
129 $text      =~ s/~\char'176\relax /g; # ~
130 $text      =~ s/|\|$/g;                   # |
131 $text      =~ s/_/\\_/g;                   # _
132 $text      =~ s/#/\char'043\relax /g; # #
133 $text      =~ s/%/\\%/g;                   # %
134 $text      =~ s/</\$/g;                   # <
135 $text      =~ s/>/\$/g;                   # >
136 #-----reset the backslash
137 $text      =~ s/dollarbackslashadolla/\$\backslash$/g;

138 return $text;
139 }
```

test.tex

5

test.tex

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %% DEFINITIONS
3  % Ed Cashin's "supersmash" macro, based on smash, TeXbook, p. 360
4  \def\supersmash#1{\relax\setbox0=\hbox{#1}
5     \ht0=0in \dp0=0in \wd0=0in \box0 }

6  \def\backgroundfig{test-static.1}
7  \def\foregroundfig{test-dyn.1}

8  \newdimen\tempdim
9  \tempdim=11in
10 \advance\tempdim by -12mm
11 \hsize \tempdim
12 \tempdim=8.5in\advance\tempdim by -5mm
13 \vsize \tempdim
14 \hoffset -.75in
15 \voffset -.85in

16                                     % here we test to see whether we're doing
17                                     % postscript or pdf:
18 \ifx\pdfoutput\undefined
19     \input epsf
20     \special{papersize=11in,8.5in}
21 \else
22     \input supp-pdf                 % mp support for pdftex
23     \pdfpageheight=8.5in           % landscape letter pagesize
24     \pdfpagewidth=11in
25 \fi

26 \nopagenumbers
27 \parindent=0in

28 % begin figure inclusion -----
29 \ifx\pdfoutput\undefined
30     \hbox{\supersmash{\epsfbox{\backgroundfig}}}%
31     \epsfbox{\foregroundfig}}
32 \else
33     \hbox{\supersmash{\convertMPToPDF {\backgroundfig} {1}{1}}}%
34     \convertMPToPDF {\foregroundfig} {1}{1}}
35 \fi
36 \vfil
37 % end figure inclusion -----

38 \bye
```

