



# METATYPE1

Bogusław Jackowski, Bacho $\TeX$ , 29 IV–3 V 2015

excerpted from the presentation delivered at the Euro $\TeX$  meeting in Kerkrade, 23–27 IX 2001

<http://www.ntg.nl/eurotex/JackowskiMT.pdf>

## POSTSCRIPT TYPE 1 FONT: WHAT IS IT?

METATYPE 1

## POSTSCRIPT TYPE 1 FONT: WHAT IS IT?

- **Metric file:** AFM (ASCII)  
*glyph names, dimensions, kerns, ligatures*

METATYPE 1

## POSTSCRIPT TYPE 1 FONT: WHAT IS IT?

- **Metric file:** AFM (ASCII)  
*glyph names, dimensions, kerns, ligatures*
- **Glyph file:** PFB (binary) or PFA (ASCII)  
*glyph shapes (outlines, no pens, no overlapping),  
encoding, hinting, some metric data*

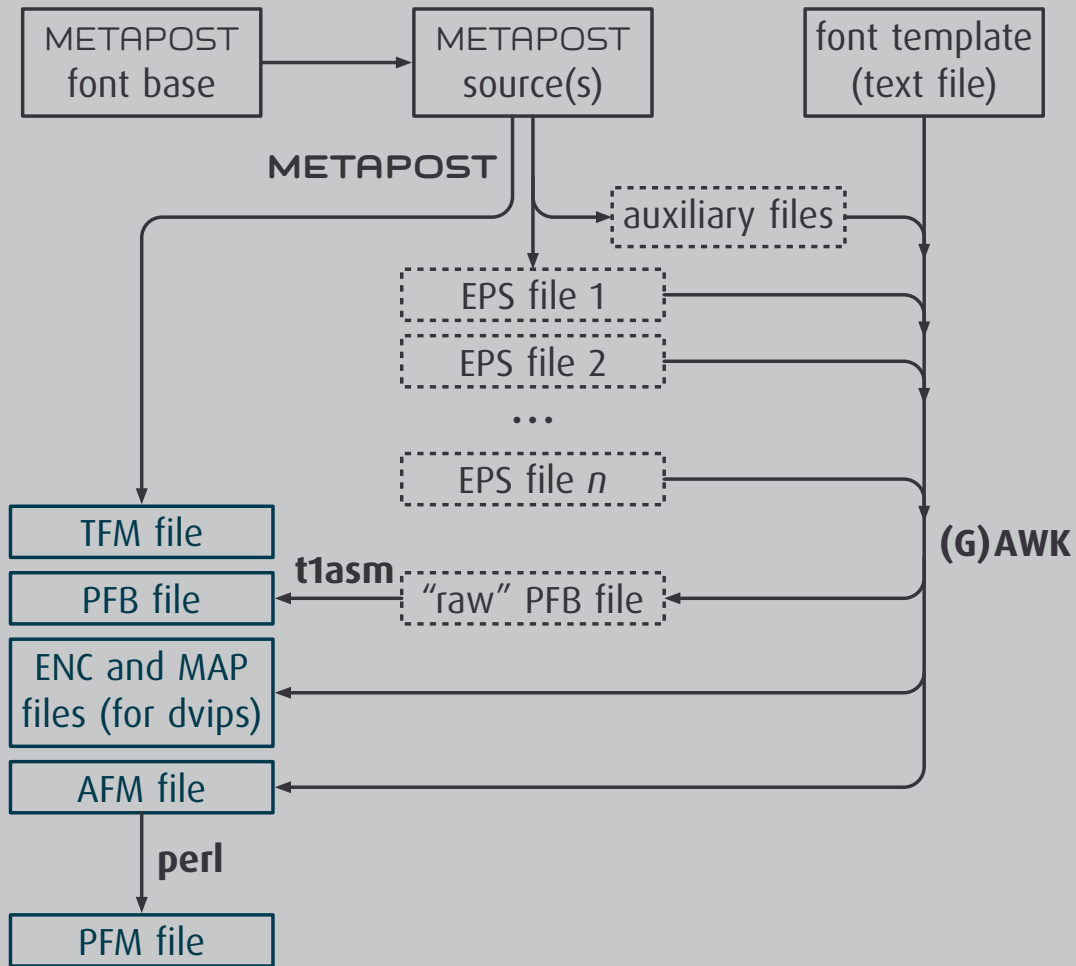
METATYPE 1

## POSTSCRIPT TYPE 1 FONT: WHAT IS IT?

- **Metric file:** AFM (ASCII)  
*glyph names, dimensions, kerns, ligatures*
- **Glyph file:** PFB (binary) or PFA (ASCII)  
*glyph shapes (outlines, no pens, no overlapping),  
encoding, hinting, some metric data*
- **Windows (“printer”) metric file:** PFM (binary)  
*contains nearly the same information as AFM,  
except glyph names*

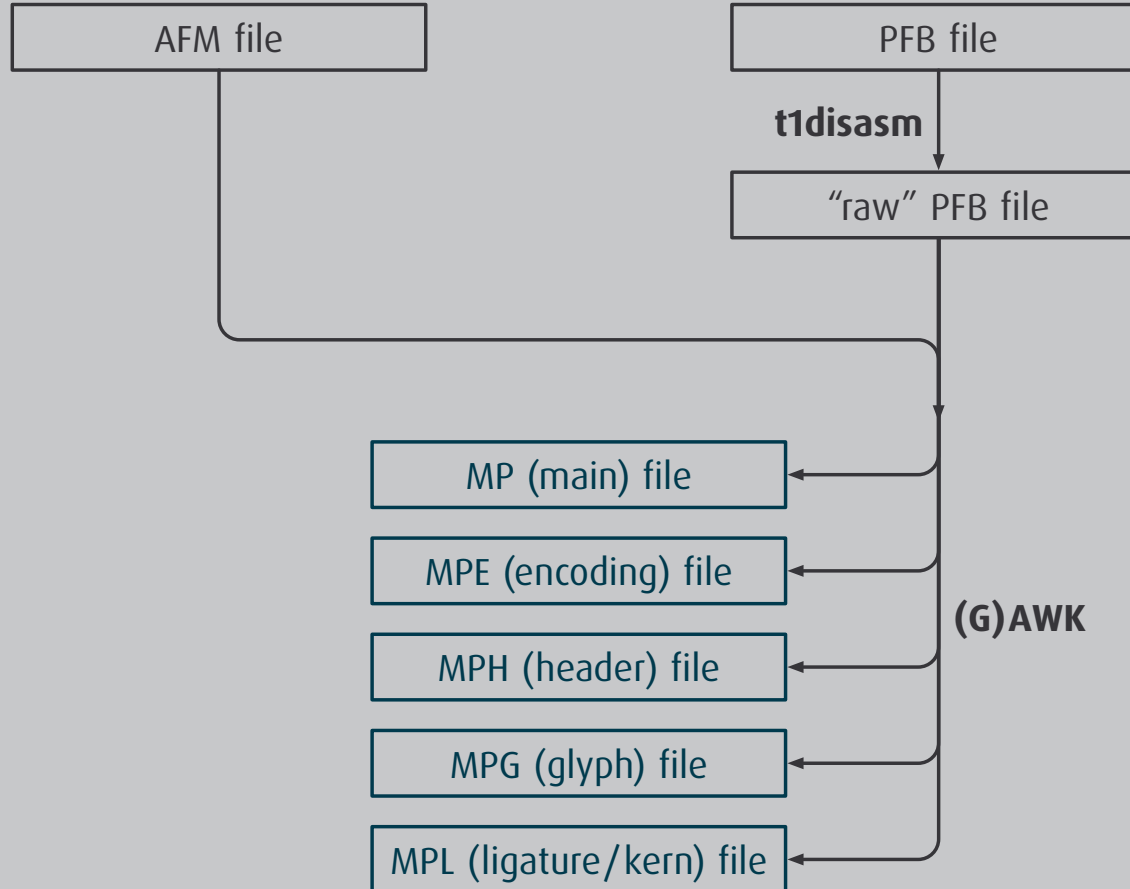
METATYPE 1

# THE SCHEME OF THE ENGINE



# METATYPE 1

## THE SCHEME OF THE "REVERSE" ENGINE



# METATYPE 1



## THE OUTPUT OF THE "REVERSE" ENGINE:

main (driver)

encoding

header

glyph

ligature/kern

```
input fontbase;  
input logo10.mpe;  
beginfont  
maybeinput "logo10.mph";  
maybeinput "logo10.mpg";  
maybeinput "logo10.mp1";  
endfont
```

METATYPE 1

## THE OUTPUT OF THE "REVERSE" ENGINE:

main (driver)      encoding      header      glyph      ligature/kern

```
% ENCODING  
encode("space", 32);  
encode("A", 65);  
encode("E", 69);  
encode("F", 70);  
encode("M", 77);  
encode("N", 78);  
encode("O", 79);  
encode("P", 80);  
encode("S", 83);  
encode("T", 84);  
endinput
```

METATYPE 1

## THE OUTPUT OF THE "REVERSE" ENGINE:

main (driver)      encoding      header      glyph      ligature/kern

```
% FONT INFORMATIONS
pf_info_familyname "MFlogo";
pf_info_weight "Normal";
pf_info_fontname "MFlogo10-Regular";
pf_info_version "1.00";
pf_info_author "Author: □Donald□E.□Knuth";
pf_info_encoding "AEFMNOPST□only";
pf_info_italicangle 0;
pf_info_underline -198, 66;
pf_info_fixedpitch false;
pf_info_adl 750, 250, 0;
pf_info_forcebold false;
blue_fuzz := 0;
blue_scale := 0.04546;
blue_shift := 7;
pf_info_overshoots(0, -11.1084), (600.0061, 11.1084);
```

# METATYPE 1

## THE OUTPUT OF THE "REVERSE" ENGINE:

main (driver)      encoding      header      glyph      ligature/kern

```
pf_info_designsize 10;
pf_info_space 266.6626, 133.3313, 88.88245;
pf_info_quad 800.00305;
pf_info_pfm "MFlogo10", 0, 0;
pf_info_capheight 600.0061;
pf_info_xheight 0;
pf_info_creationdate;
italic_shift := 0;

% INTRODUCE CHARS
standard_introduce("space");
standard_introduce("A");
standard_introduce("E");
standard_introduce("F");
standard_introduce("M");
standard_introduce("N");
standard_introduce("O");
standard_introduce("P");
```

# METATYPE 1

## THE OUTPUT OF THE "REVERSE" ENGINE:

main (driver)      encoding      header      glyph      ligature/kern

```
standard_introduce("S");
standard_introduce("T");
standard_introduce("nbspace");

% METRICS
wd.space = 267; ht.space = 0; dp.space = 0;
wd.A = 666.67175; ht.A = 611; dp.A = -11;
wd.E = 622.2229; ht.E = 600; dp.E = 0;
wd.F = 622.2229; ht.F = 600; dp.F = -11;
wd.M = 800.00305; ht.M = 611; dp.M = -11;
wd.N = 666.67175; ht.N = 611; dp.N = -11;
wd.O = 666.67175; ht.O = 611; dp.O = -11;
wd.P = 622.2229; ht.P = 600; dp.P = -11;
wd.S = 622.2229; ht.S = 600; dp.S = 0;
wd.T = 577.77405; ht.T = 600; dp.T = -11;
wd.nspace = 267; ht.nspace = 0; dp.nspace = 0;
endinput
```

# METATYPE 1

## THE OUTPUT OF THE “REVERSE” ENGINE:

main (driver)      encoding      header      glyph      ligature/kern

```
beginglyph(T);
  save p; path p[];

  z0 0 = (578, 570); z0 0a = (578, 587); z0 1b = (563, 600);
  z0 1 = (545, 600);
  z0 2 = (33, 600); z0 2a = (15, 600); z0 3b = (0, 587);
  z0 3 = (0, 570); z0 3a = (0, 553); z0 4b = (15, 540);
  z0 4 = (33, 540);
  z0 5 = (256, 540);
  z0 6 = (256, 19); z0 6a = (256, 2); z0 7b = (271, -11);
  z0 7 = (289, -11); z0 7a = (307, -11); z0 8b = (322, 2);
  z0 8 = (322, 19);
  z0 9 = (322, 540);
  z0 10 = (545, 540); z0 10a = (563, 540); z0 11b = (578, 553);
  z0 11 = (578, 570);
  p0 = compose_pathz0(11);

  if turningnumber p0 > 0: Fill else: unFill fi p0;
```

# METATYPE 1

## THE OUTPUT OF THE "REVERSE" ENGINE:

main (driver)    encoding    header    glyph    ligature/kern

```
fix_hstem(60)(p0)candidate_list(y)(540, 600);  
fix_vstem(66)(p0)candidate_list(x)(256, 322);  
standard_exact_hsbw("T");  
endglyph;
```

METATYPE1

# THE OUTPUT OF THE "REVERSE" ENGINE:

main (driver)

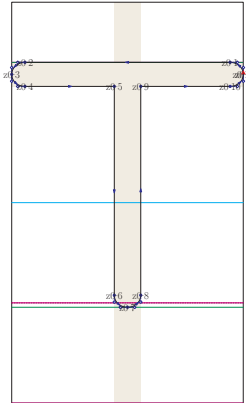
encoding

header

glyph

ligature/kern

Construction of the character T: | Konstrukcja znaku T:



```
beginlyph(T);
save p; path p[];
z0_0 = (578, 570); z0_0a = (578, 587); z0_1b = (563, 600);
z0_1 = (545, 600);
z0_2 = (33, 600); z0_2a = (15, 600); z0_3b = (0, 587);
z0_3 = (0, 570); z0_3a = (0, 553); z0_4b = (15, 540);
z0_4 = (33, 540);
z0_5 = (256, 540);
z0_6 = (256, 19); z0_6a = (256, 2); z0_7b = (271, -11);
z0_7 = (289, -11); z0_7a = (307, -11); z0_8b = (322, 2);
z0_8 = (322, 19);
z0_9 = (322, 540);
z0_10 = (545, 540); z0_10a = (563, 540); z0_11b = (578, 553);
z0_11 = (578, 570);
p0 = compose_path_z0(11);
if turningnumber p0 > 0: Fill else: unFill fi p0;
fix_hstem(60)(p0) candidate_list(y)(540, 600);
fix_vstem(66)(p0) candidate_list(x)(256, 322);
standard_exact_hsbw("T");
endglyph;
```

# METATYPE1



## THE OUTPUT OF THE "REVERSE" ENGINE:

main (driver)    encoding    header    glyph    ligature/kern

```
% KERN PAIRS  
LK("F")  
    KP("0", -44.44885)  
KL;  
LK("P")  
    KP("0", 44.44885)  
KL;  
LK("T")  
    KP("A", -22.22443)  
KL;  
endinput
```

# METATYPE1

AN EXAMPLE:  
the letter "T" from D. E. Knuth's logo font

```
beginlogochar("T", 13);  
italcorr ht# * slant + .5u#;  
if .5w <> good.x .5w: change_width; fi  
lft  $x_1 = -eps$ ;  $x_2 = w - x_1$ ;  $x_3 = x_4 = .5w$ ;  
 $y_1 = y_2 = y_3$ ; top  $y_1 = h$ ; bot  $y_4 = -o$ ;  
draw  $z_1 -- z_2$ ; draw  $z_3 -- z_4$ ;  
labels(1, 2, 3, 4);  
endchar;
```

METAFONT source

METATYPE1

## AN EXAMPLE: the letter "T" from D. E. Knuth's logo font

```
beginlogochar("T", 13);  
italcorr ht# * slant + .5u#;  
if .5w <> good.x .5w: change_width; fi  
lft  $x_1 = -eps$ ;  $x_2 = w - x_1$ ;  $x_3 = x_4 = .5w$ ;  
 $y_1 = y_2 = y_3$ ; top  $y_1 = h$ ; bot  $y_4 = -o$ ;  
draw  $z_1 -- z_2$ ; draw  $z_3 -- z_4$ ;  
labels(1, 2, 3, 4);  
endchar;
```

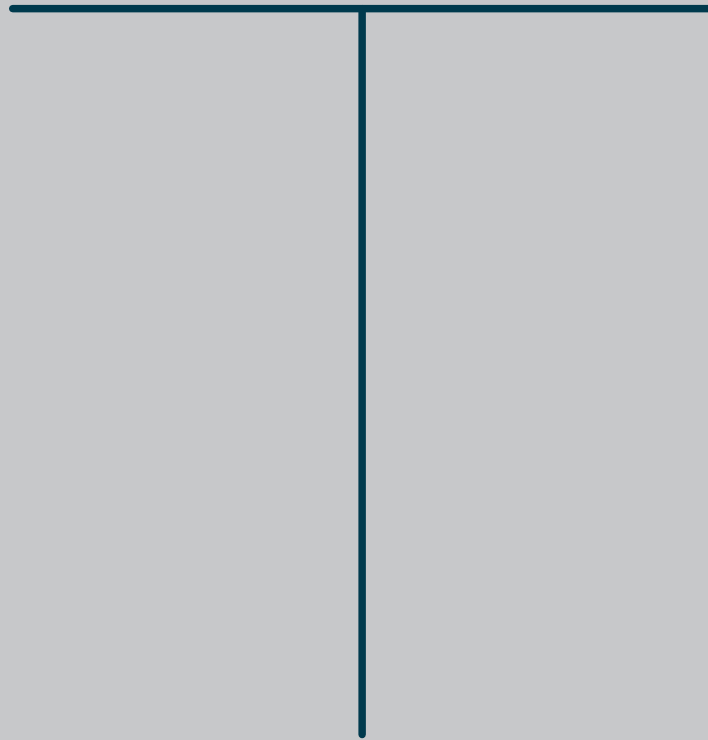
METAFONT source

```
beginlogoglyph("T", 13);  
ic.T# := ht# * slant + .5u#; if ic.T# < 0: ic.T# := 0; fi  
lft  $x_1 = 0$ ;  $x_2 = w - x_1$ ;  $x_3 = x_4 = .5w$ ;  
 $y_1 = y_2 = y_3$ ; top  $y_1 = h$ ; bot  $y_4 = -o$ ;  
find_outlines(expand_logo_stroke)( $z_1 -- z_2$ ),  
          expand_logo_stroke(butt_end(0))( $z_3 -- z_4$ ))(glyph);  
Fill glyph1; fix_vstem(px)(glyph1); fix_hstem(py)(glyph1);  
just_labels(1, 2, 3, 4);  
endlogoglyph;
```

METAPOST source

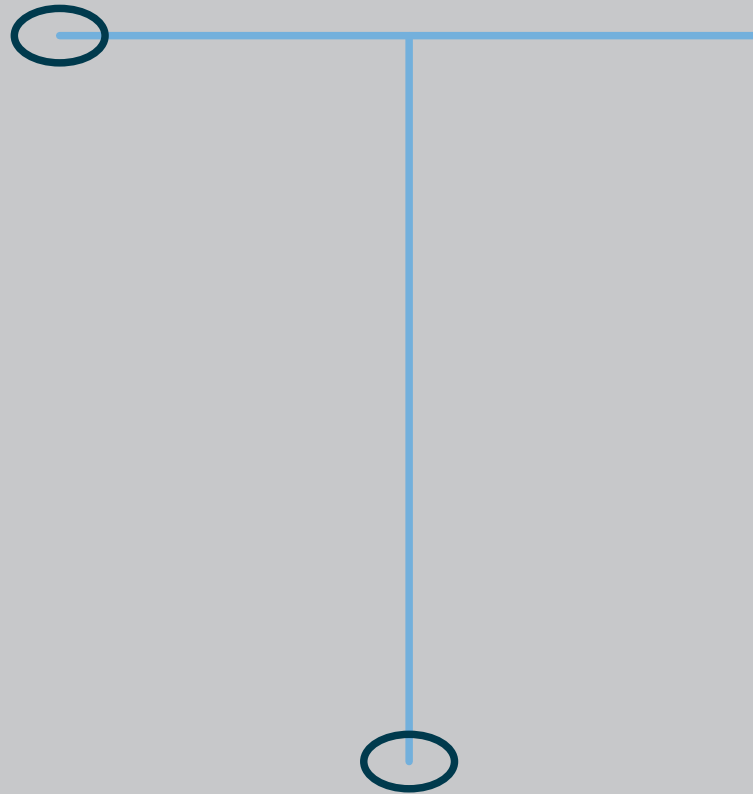
# METATYPE1

AN EXAMPLE:  
the construction of the letter "T"



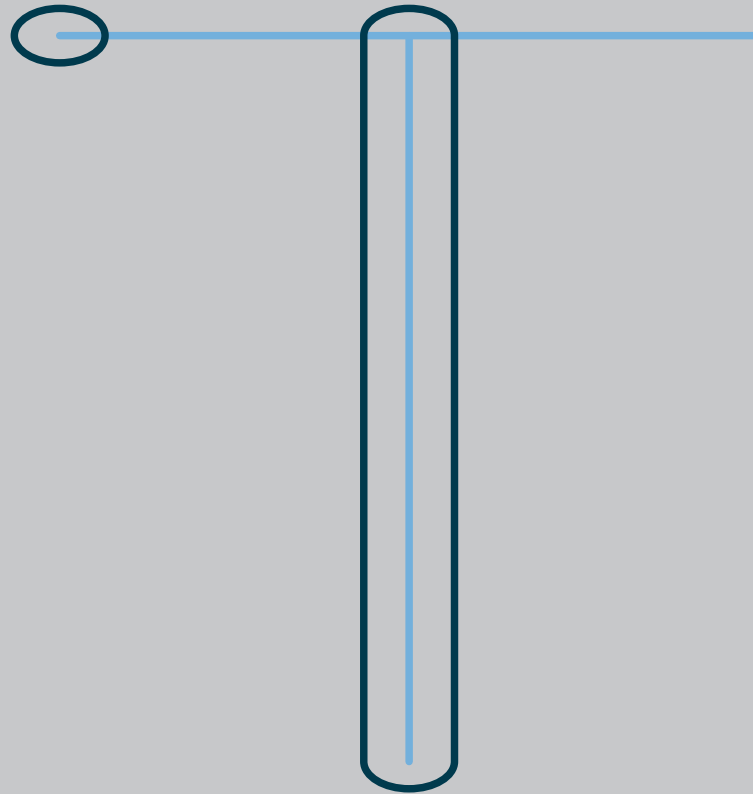
METATYPE1

AN EXAMPLE:  
the construction of the letter "T"



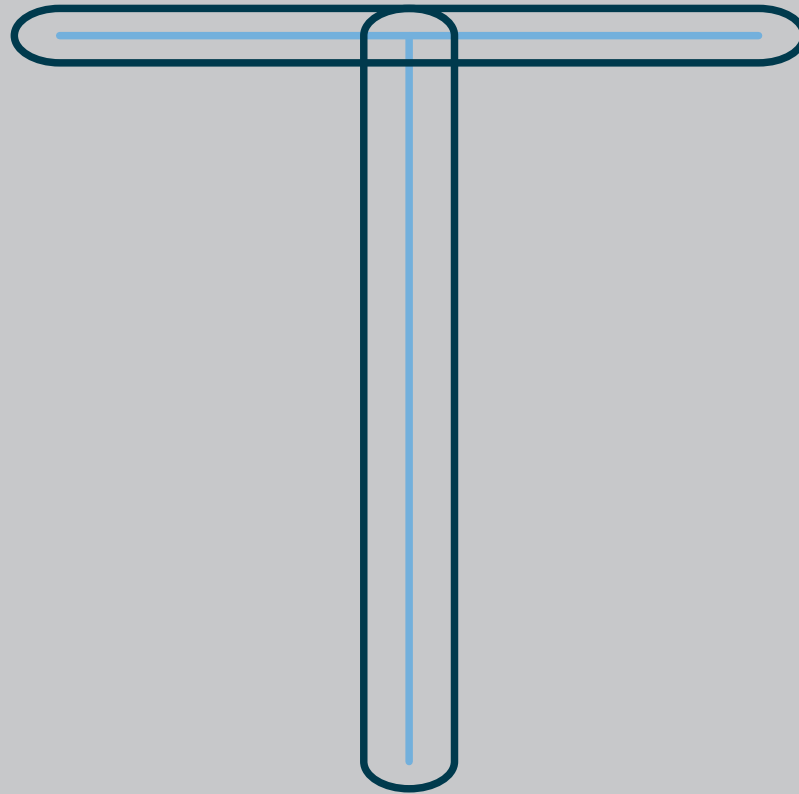
METATYPE1

AN EXAMPLE:  
the construction of the letter "T"



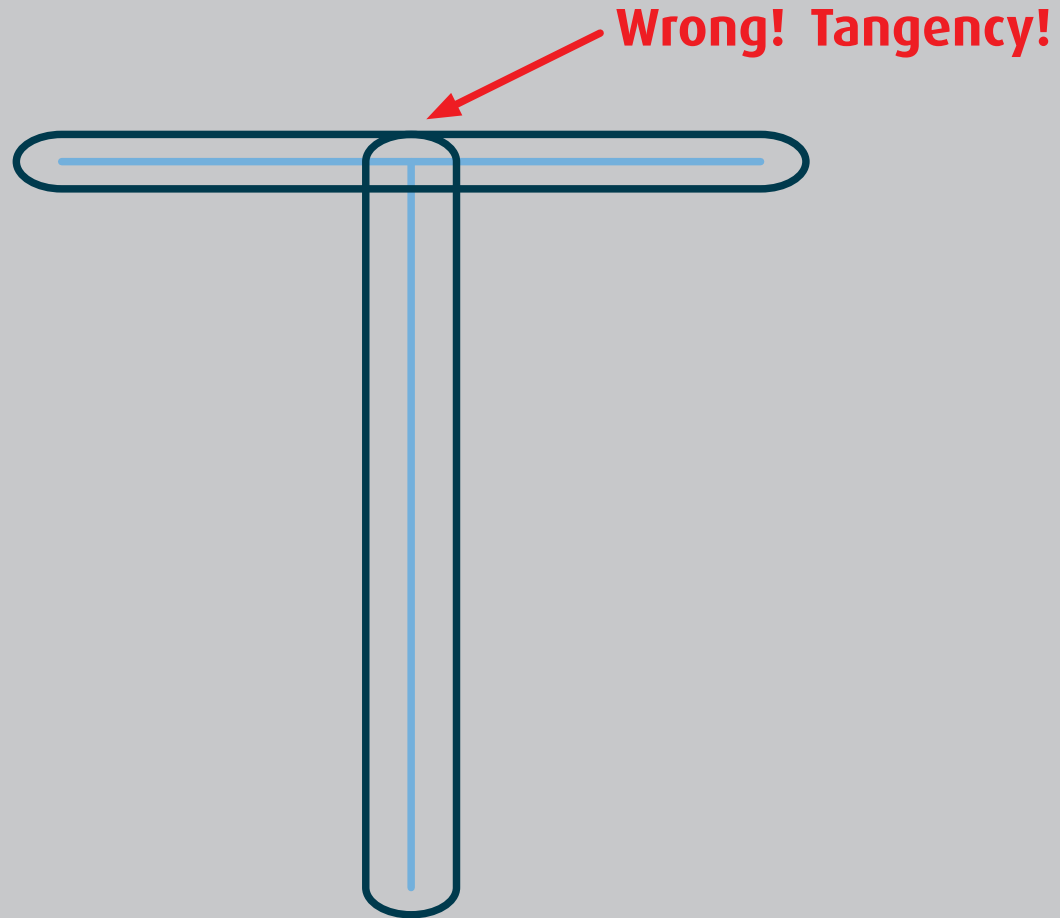
METATYPE1

AN EXAMPLE:  
the construction of the letter "T"



METATYPE1

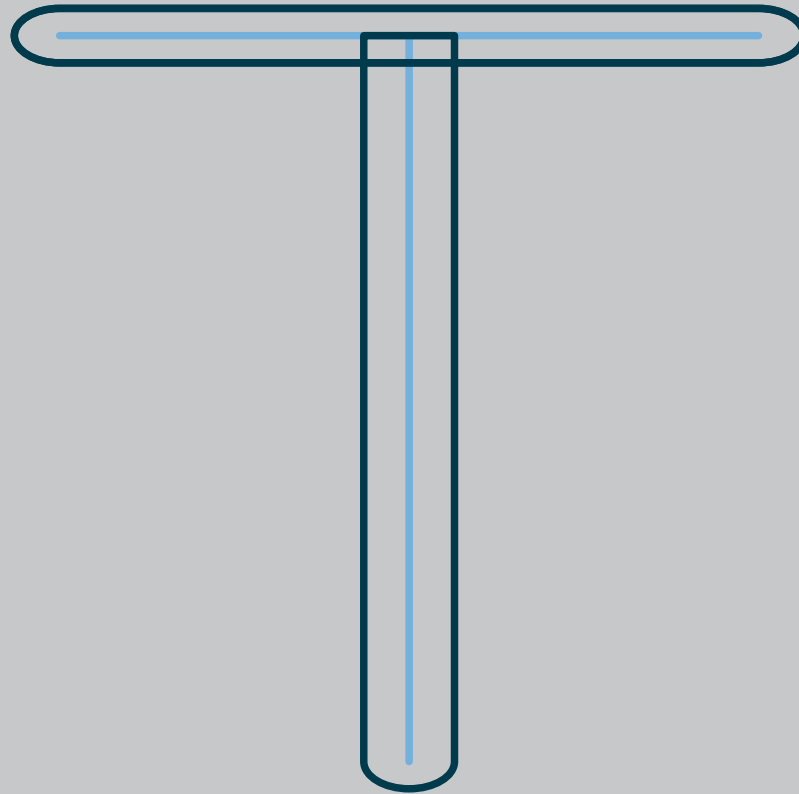
AN EXAMPLE:  
the construction of the letter "T"



METATYPE1

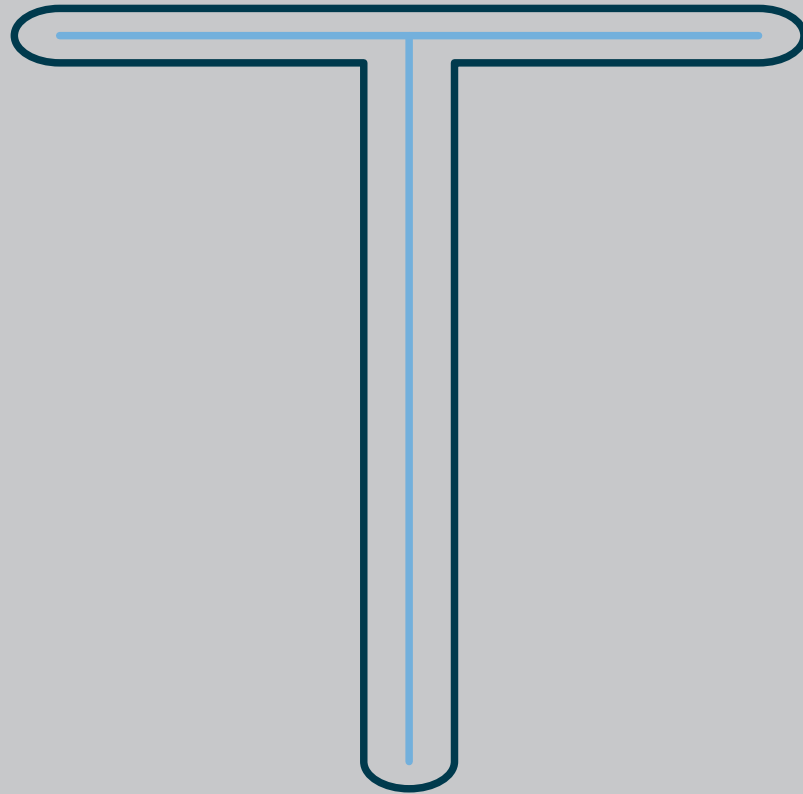


AN EXAMPLE:  
the construction of the letter "T"



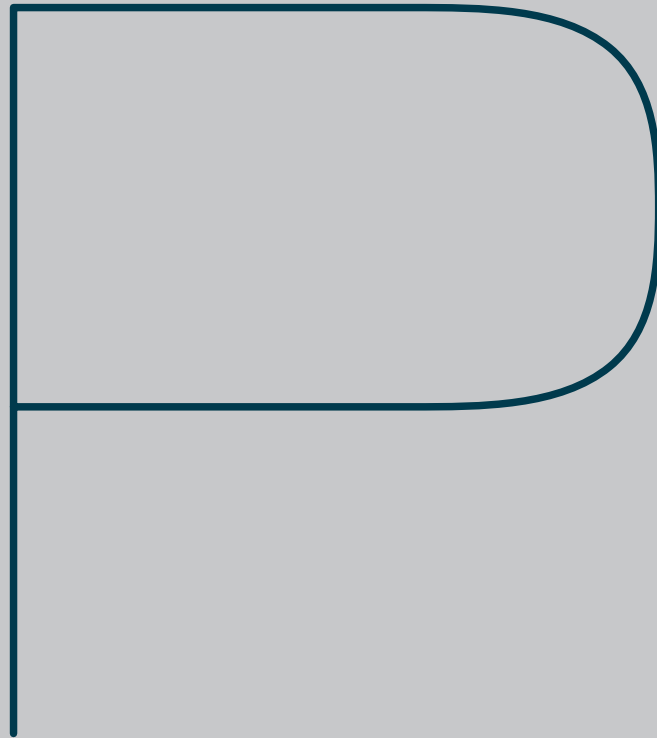
METATYPE1

AN EXAMPLE:  
the construction of the letter "T"



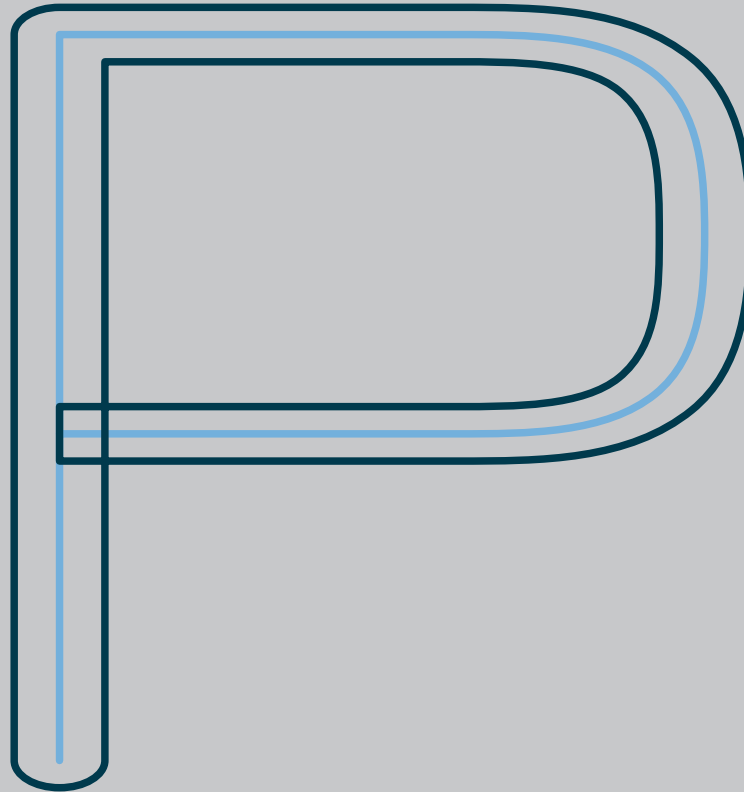
METATYPE1

AN EXAMPLE:  
the construction of the letter "P"



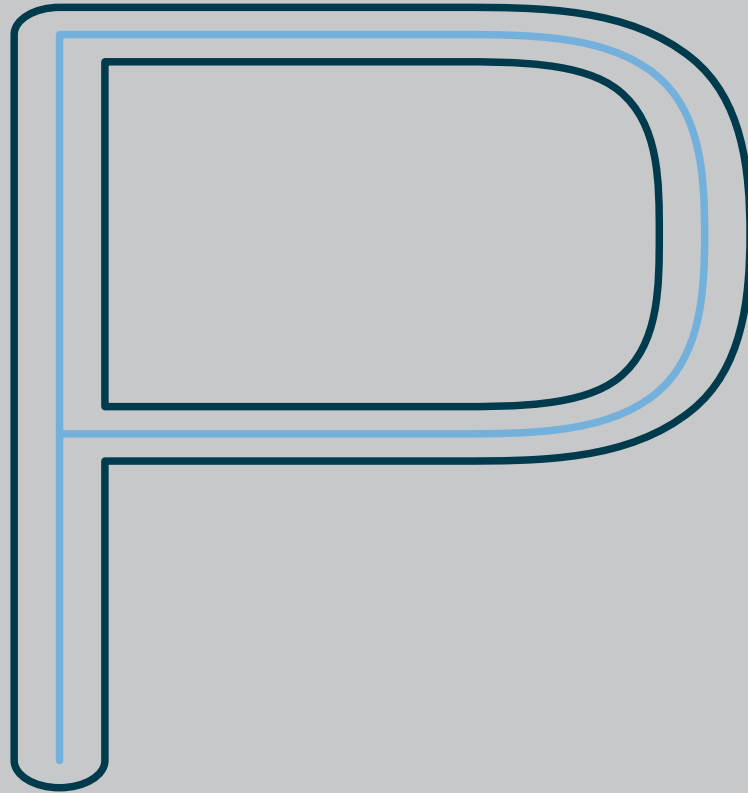
METATYPE1

AN EXAMPLE:  
the construction of the letter "P"



METATYPE1

AN EXAMPLE:  
the construction of the letter "P"



METATYPE1

# THE FINAL EXHORTATION

LET US GO FORTH NOW

and create masterpieces of digital typography  
in outline font formats

METATYPE1