

Headeren

En METAFONT-fil skal altid starte med nogle deklARATIONER. Jeg har brugt følgende:

```
mode_setup;

font_size 10pt#;
font_normal_space 5pt#;
font_normal_stretch 10pt#;

font_coding_scheme := "Test font";

h# := 5pt#;
w# := 5pt#;
d# := 2pt#;

define_pixels(h,d);
define_blacker_pixels(w);
```

Brug i L^AT_EX

For at bruge sit nye font i L^AT_EX kan man fx skrive flg.:

```
\DeclareFontFamily{U}{test}{}  
\DeclareFontShape{U}{test}{m}{n}{ <-> test10 }{}  
\newcommand\testfamily{\fontencoding{U}%  
    \fontfamily{test}\selectfont}  
\DeclareTextFontCommand{\texttest}{\testfamily}
```

Derefter kan man skrive `{\texttest abc}`.

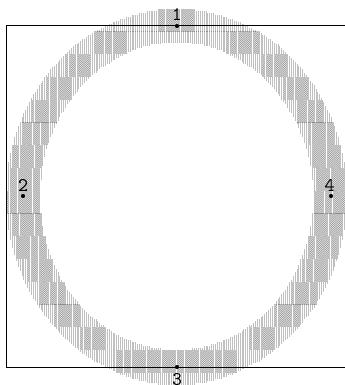
Penne

Man vælger penne v. h. a. kommandoen
pickup, efterfulgt af et penudtryk.

pencircle

En cirkel, tegnet med pencircle: O.

METAFONT output 1999.04.22:1521 Page 1 Character 97 "pencircle"

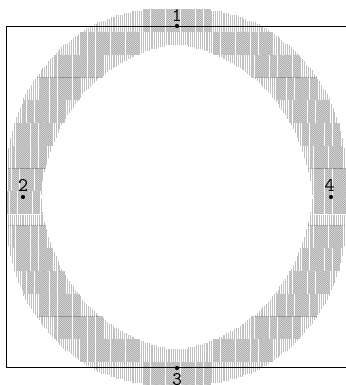


```
beginchar("a", w#, h#, 0); "pencircle";  
  pickup pencircle scaled .5pt;  
  
  x1 = x3; x2 = w - x4; x1 = .5[x2,x4]; x2 = 1/20w;  
  y1 = h; y3 = 0; y2 = y4 = .5[y1,y3];  
  
  draw z1..z2..z3..z4..cycle;  
  
  penlabels(1,2,3,4);  
endchar;
```

pensquare

En cirkel, tegnet med pensquare: O.

METAFONT output 1999.04.22:1521 Page 1 Character 98 "pensquare"



```
beginchar("b", w#, h#, 0); "pensquare";
  pickup pensquare scaled .5pt;

  x1 = x3; x2 = w - x4; x1 = .5[x2,x4]; x2 = 1/20w;
  y1 = h; y3 = 0; y2 = y4 = .5[y1,y3];

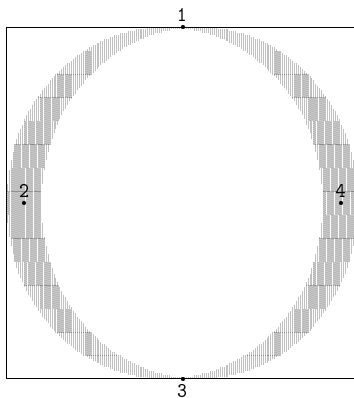
  draw z1..z2..z3..z4..cycle;

  penlabels(1,2,3,4);
endchar;
```

penrazor

En cirkel, tegnet med penrazor: ○.

METAFONT output 1999.04.22:1521 Page 1 Character 99 "penrazor"



```
beginchar("c", w#, h#, 0); "penrazor";
  pickup penrazor scaled .5pt;

  x1 = x3; x2 = w - x4; x1 = .5[x2,x4]; x2 = 1/20w;
  y1 = h; y3 = 0; y2 = y4 = .5[y1,y3];

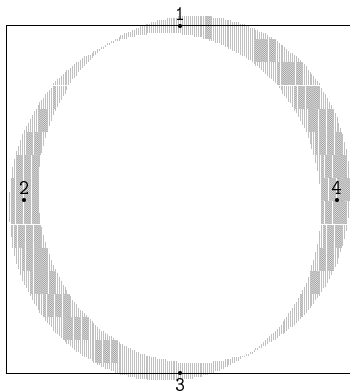
  draw z1..z2..z3..z4..cycle;

  penlabels(1,2,3,4);
endchar;
```

rotated

En cirkel, tegnet med `rotated 30`: ○.

METAFONT output 1999.04.22:1521 Page 1 Character 100 "penrazor rotated"



```
beginchar("d", w#, h#, 0); "penrazor rotated";
  pickup penrazor scaled .5pt rotated 30;

  x1 = x3; x2 = w - x4; x1 = .5[x2,x4]; x2 = 1/20w;
  y1 = h; y3 = 0; y2 = y4 = .5[y1,y3];

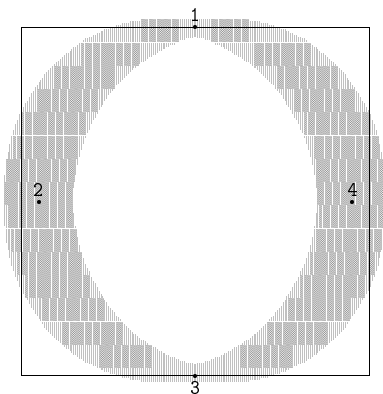
  draw z1..z2..z3..z4..cycle;

  penlabels(1,2,3,4);
endchar;
```

xscaled yscaled

En cirkel, tegnet med pensquare: O.

METAFONT output 1999.04.22:1521 Page 1 Character 101 "pensquare xscaled yscaled"



```
beginchar("e", w#, h#, 0); "pensquare xscaled yscaled";
  pickup pensquare xscaled 1pt yscaled .2pt;

  x1 = x3; x2 = w - x4; x1 = .5[x2,x4]; x2 = 1/20w;
  y1 = h; y3 = 0; y2 = y4 = .5[y1,y3];

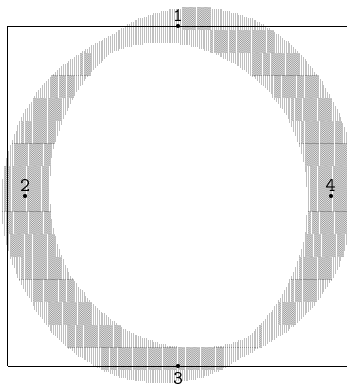
  draw z1..z2..z3..z4..cycle;

  penlabels(1,2,3,4);
endchar;
```

En avanceret pen

En cirkel, tegnet med en flot pen: O.

METAFONT output 1999.04.22:1521 Page 1 Character 102 "Flot pen"



```
beginchar("f", w#, h#, 0); "Flot pen";
  pickup pensquare xscaled .7pt yscaled .2pt rotated 30;

  x1 = x3; x2 = w - x4; x1 = .5[x2,x4]; x2 = 1/20w;
  y1 = h; y3 = 0; y2 = y4 = .5[y1,y3];

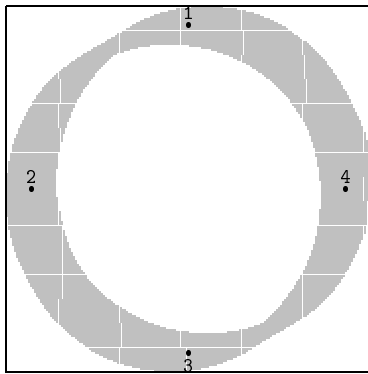
  draw z1..z2..z3..z4..cycle;

  penlabels(1,2,3,4);
endchar;
```

Placering af punkter

Man kan sørge for, at punkter ikke rammer udenfor et område, hvis man bruger `top`, `bot`, `lft` og `rt`.

METAFONT output 1999.04.22:1521 Page 1 Character 103 "top bot lft rt"



```
beginchar("g", w#, h#, 0); "top bot lft rt";
  pickup pensquare xscaled .7pt yscaled .2pt rotated 30;

  x1 = x3; rt x4 = w; x1 = .5[x2,x4]; lft x2 = 0;
  top y1 = h; bot y3 = 0; y2 = y4 = .5[y1,y3];

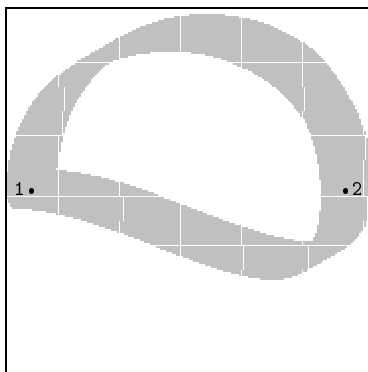
  draw z1..z2..z3..z4..cycle;

  penlabels(1,2,3,4);
endchar;
```

Retninger

Man kan ændre retning på streger med `up`,
`down`, `right` og `left`.

METAFONT output 1999.04.22:1521 Page 1 Character 107 "up down right left"



```
beginchar("k", w#, h#, 0); "up down right left";
  pickup pensquare xscaled .7pt yscaled .2pt rotated 30;

  lft x1 = 0; rt x2 = w; y1 = y2 = .5h;

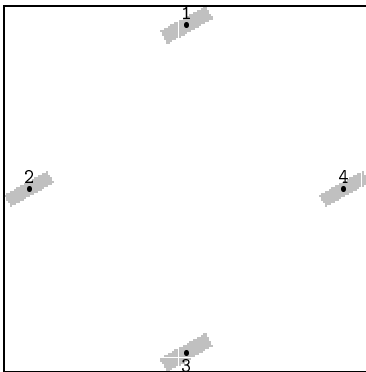
  draw z1{up}..{down}z2;
  draw z1{right}..{up}z2;

  penlabels(1,2);
endchar;
```

t tegne prikker

Prikker tegnes med drawdot.

METAFONT output 1999.04.22:1521 Page 1 Character 104 "drawdot"

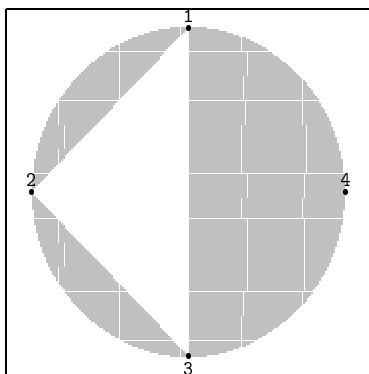


```
beginchar("h", w#, h#, 0); "drawdot";  
  pickup pensquare xscaled .7pt yscaled .2pt rotated 30;  
  
  x1 = x3; rt x4 = w; x1 = .5[x2,x4]; lft x2 = 0;  
  top y1 = h; bot y3 = 0; y2 = y4 = .5[y1,y3];  
  
  drawdot z1; drawdot z2; drawdot z3; drawdot z4;  
  
  penlabels(1,2,3,4);  
endchar;
```

Udfyldning

Man kan fylde områder med `fill` og affylde dem med `unfill`: ●.

METAFONT output 1999.04.22:1521 Page 1 Character 108 "drawdot"



```
beginchar("l", w#, h#, 0); "drawdot";
  pickup pensquare xscaled .7pt yscaled .2pt rotated 30;

  x1 = x3; rt x4 = w; x1 = .5[x2,x4]; lft x2 = 0;
  top y1 = h; bot y3 = 0; y2 = y4 = .5[y1,y3];

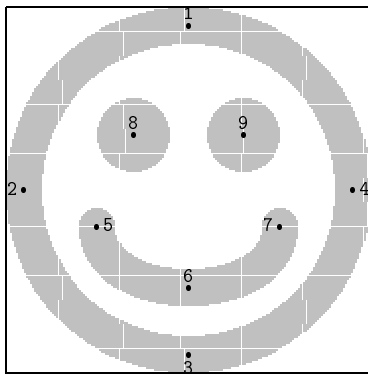
  fill z1..z2..z3..z4..cycle;
  unfill z1--z2--z3--cycle;

  penlabels(1,2,3,4);
endchar;
```

Et par eksempler

Lad os starte med at lave en smiley: ☺.

METAFONT output 1999.04.22:1521 Page 1 Character 105 "Smiley"



```
beginchar("i", w#, h#, 0); "Smiley";
  pickup pencircle scaled .5pt;

  x1 = x3; rt x4 = w; x1 = .5[x2,x4]; lft x2 = 0;
  top y1 = h; bot y3 = 0; y2 = y4 = .5[y1,y3];

  draw z1..z2..z3..z4..cycle;

  x5 = w - x7; x5 = x2 + 1pt; x6 = .5[x5,x7];
  y5 = y7; bot y6 = y3 + 2/3pt; bot y5 = 1/3[y3,y1];

  draw z5{down}..z6..{up}z7;

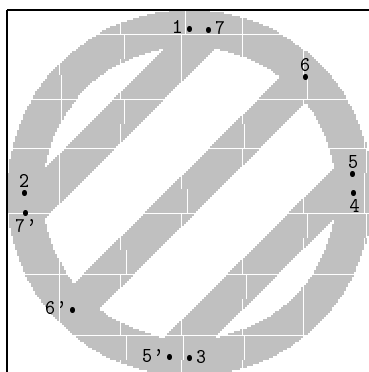
  pickup pencircle scaled 1pt;
  lft x8 = w - rt x9 = x5;
  y8 = y9 = 2/3[y3,y1];

  drawdot z8; drawdot z9;

  penlabels(1,2,3,4,5,6,7,8,9);
endchar;
```

Og hvad med en skraveret cirkel: ø?

METAFONT output 1999.04.22:1521 Page 1 Character 106 "Skraveret cirkel"



```
beginchar("j", w#, h#, 0); "Skraveret cirkel";
  pickup pencircle scaled .5pt;

  x1 = x3; rt x4 = w; x1 = .5[x2,x4]; lft x2 = 0;
  top y1 = h; bot y3 = 0; y2 = y4 = .5[y1,y3];

  draw z1..z2..z3..z4..cycle;

  z10 = (x1, y2);

  z6 = dir(45) xscaled (x4-x10) yscaled (y1-y10) shifted z10;
  z6' = dir(45 + 180) xscaled (x4-x10) yscaled (y1-y10) shifted z10;

  z5 = dir(sqrt 45) xscaled (x4-x10) yscaled (y1-y10) shifted z10;
  z5' = dir(225 + (45-sqrt 45)) xscaled (x4-x10) yscaled (y1-y10) shifted z10;

  z7 = dir(45 + (45-sqrt 45)) xscaled (x4-x10) yscaled (y1-y10) shifted z10;
  z7' = dir(180 + sqrt(45)) xscaled (x4-x10) yscaled (y1-y10) shifted z10;

  draw z6--z6';   draw z5--z5';   draw z7--z7';

  penlabels(1,2,3,4,6,6',5,5',7,7');

endchar;
```

t køre METAFONT

Der er to hovedmåder at køre METAFONT på.

Korrekturudskift

```
mf "\mode=proof; input font.mf"  
gftodvi font.2602gf
```

Resultatet er font.dvi.

Font

```
mf "\mode=localfont; input font.mf"  
gftopk font.600gf
```

Resultatet er font.tfm og font.600pk.

Computer Modern

Knuths *Computer Modern*-font er i meget høj grad parametriseret.

Her er parametrene for standardfontet (cmr10):

```
% Computer Modern Roman 10 point
if unknown cmbase: input cmbase fi

font_identifier:="CMR"; font_size 10pt#;

u#:=20/36pt#;      % unit width
width_adj#:=0pt#;  % width adjustment for certain characters
serif_fit#:=0pt#;  % extra sidebar near lowercase serifs
cap_serif_fit#:=5/36pt#; % extra sidebar near uppercase serifs
letter_fit#:=0pt#; % extra space added to all sidebars

body_height#:=270/36pt#; % height of tallest characters
asc_height#:=250/36pt#;  % height of lowercase ascenders
cap_height#:=246/36pt#;  % height of caps
fig_height#:=232/36pt#;  % height of numerals
x_height#:=155/36pt#;    % height of lowercase without ascenders
math_axis#:=90/36pt#;   % axis of symmetry for math symbols
bar_height#:=87/36pt#;   % height of crossbar in lowercase e
comma_depth#:=70/36pt#; % depth of comma below baseline
desc_depth#:=70/36pt#;  % depth of lowercase descenders

crisp#:=0pt#;          % diameter of serif corners
tiny#:=8/36pt#;        % diameter of rounded corners
fine#:=7/36pt#;        % diameter of sharply rounded corners
thin_join#:=7/36pt#;   % width of extrafine details
hair#:=9/36pt#;        % lowercase hairline breadth
stem#:=25/36pt#;       % lowercase stem breadth
curve#:=30/36pt#;      % lowercase curve breadth
ess#:=27/36pt#;        % breadth in middle of lowercase s
flare#:=33/36pt#;      % diameter of bulbs or breadth of terminals
dot_size#:=38/36pt#;   % diameter of dots
cap_hair#:=11/36pt#;   % uppercase hairline breadth
cap_stem#:=32/36pt#;   % uppercase stem breadth
```

```

cap_curve#:=37/36pt#;    % uppercase curve breadth
cap_ess#:=35/36pt#;    % breadth in middle of uppercase s
rule_thickness#:=.4pt#;    % thickness of lines in math symbols

dish#:=1/36pt#;        % amount erased at top or bottom of serifs
bracket#:=20/36pt#;    % vertical distance from serif base to tangent
jut#:=28/36pt#;        % protrusion of lowercase serifs
cap_jut#:=37/36pt#;    % protrusion of uppercase serifs
beak_jut#:=10/36pt#;   % horizontal protrusion of beak serifs
beak#:=70/36pt#;      % vertical protrusion of beak serifs
vair#:=8/36pt#;       % vertical diameter of hairlines
notch_cut#:=10pt#;    % maximum breadth above or below notches
bar#:=11/36pt#;       % lowercase bar thickness
slab#:=11/36pt#;     % serif and arm thickness
cap_bar#:=11/36pt#;   % uppercase bar thickness
cap_band#:=11/36pt#;  % uppercase thickness above/below lobes
cap_notch_cut#:=10pt#; % max breadth above/below uppercase notches
serif_drop#:=4/36pt#; % vertical drop of sloped serifs
stem_corr#:=1/36pt#;  % for small refinements of stem breadth
vair_corr#:=1/36pt#;  % for small refinements of hairline height
apex_corr#:=0pt#;    % extra width at diagonal junctions

o#:=8/36pt#;         % amount of overshoot for curves
apex_o#:=8/36pt#;   % amount of overshoot for diagonal junctions

slant:=0;           % tilt ratio  $(\Delta x / \Delta y)$ 
fudge:=1;          % factor applied to weights of heavy characters
math_spread:=0;    % extra openness of math symbols
superiness:=1/sqrt2; % parameter for superellipses
superpull:=1/6;    % extra openness inside bowls
beak_darkness:=11/30; % fraction of triangle inside beak serifs
ligs:=2;          % level of ligatures to be included

square_dots:=false; % should dots be square?
hefty:=false;      % should we try hard not to be overweight?
serifs:=true;     % should serifs and bulbs be attached?
monospace:=false; % should all characters have the same width?
variant_g:=false; % should an italic-style g be used?
low_asterisk:=false; % should the asterisk be centered at the axis?
math_fitting:=false; % should math-mode spacing be used?

generate_roman    % switch to the driver file

```

Computer Modern

Here are parameters for Computer Concrete (ccr10):

```
% Concrete Roman 10 point
if unknown cmbase: input cmbase fi

font_identifier:="CCR"; font_size 10pt#;

u#: =20/36pt#;           % unit width
width_adj#: =0pt#;      % width adjustment for certain characters
serif_fit#: =1/36pt#;   % extra sidebar near lowercase serifs
cap_serif_fit#: =3/36pt#; % extra sidebar near uppercase serifs
letter_fit#: =0pt#;     % extra space added to all sidebars

body_height#: =270/36pt#; % height of tallest characters
asc_height#: =250/36pt#; % height of lowercase ascenders
cap_height#: =246/36pt#; % height of caps
fig_height#: =232/36pt#; % height of numerals
x_height#: =165/36pt#; % height of lowercase without ascenders
math_axis#: =90/36pt#; % axis of symmetry for math symbols
bar_height#: =92/36pt#; % height of crossbar in lowercase e
comma_depth#: =70/36pt#; % depth of comma below baseline
desc_depth#: =70/36pt#; % depth of lowercase descenders

crisp#: =0pt#;          % diameter of serif corners
tiny#: =11/36pt#;      % diameter of rounded corners
fine#: =6/36pt#;       % diameter of sharply rounded corners
thin_join#: =17/36pt#; % width of extrafine details
hair#: =21/36pt#;      % lowercase hairline breadth
stem#: =25/36pt#;      % lowercase stem breadth
curve#: =27/36pt#;     % lowercase curve breadth
ess#: =25/36pt#;       % breadth in middle of lowercase s
flare#: =29/36pt#;     % diameter of bulbs or breadth of terminals
dot_size#: =38/36pt#; % diameter of dots
cap_hair#: =21/36pt#; % uppercase hairline breadth
cap_stem#: =27/36pt#; % uppercase stem breadth
cap_curve#: =28/36pt#; % uppercase curve breadth
cap_ess#: =27/36pt#; % breadth in middle of uppercase s
rule_thickness#: =.4pt#; % thickness of lines in math symbols
```

```

dish#:=0/36pt#;           % amount erased at top or bottom of serifs
bracket#:=5/36pt#;       % vertical distance from serif base to tangent
jut#:=30/36pt#;          % protrusion of lowercase serifs
cap_jut#:=32/36pt#;      % protrusion of uppercase serifs
beak_jut#:=10/36pt#;     % horizontal protrusion of beak serifs
beak#:=70/36pt#;         % vertical protrusion of beak serifs
vair#:=21/36pt#;         % vertical diameter of hairlines
notch_cut#:=30/36pt#;    % maximum breadth above or below notches
bar#:=21/36pt#;          % lowercase bar thickness
slab#:=21/36pt#;         % serif and arm thickness
cap_bar#:=21/36pt#;      % uppercase bar thickness
cap_band#:=21/36pt#;     % uppercase thickness above/below lobes
cap_notch_cut#:=1pt#;    % max breadth above/below uppercase notches
serif_drop#:=5/36pt#;    % vertical drop of sloped serifs
stem_corr#:=1/36pt#;     % for small refinements of stem breadth
vair_corr#:=1/36pt#;     % for small refinements of hairline height
apex_corr#:=0pt#;        % extra width at diagonal junctions

o#:=4/36pt#;             % amount of overshoot for curves
apex_o#:=3/36pt#;        % amount of overshoot for diagonal junctions

slant:=0;                % tilt ratio  $(\Delta x / \Delta y)$ 
fudge:=.95;              % factor applied to weights of heavy characters
math_spread:=0;          % extra openness of math symbols
superness:=8/11;         % parameter for superellipses
superpull:=1/15;         % extra openness inside bowls
beak_darkness:=4/30;     % fraction of triangle inside beak serifs
ligs:=2;                  % level of ligatures to be included

square_dots:=false;      % should dots be square?
hefty:=false;            % should we try hard not to be overweight?
serifs:=true;            % should serifs and bulbs be attached?
monospace:=false;        % should all characters have the same width?
variant_g:=false;        % should an italic-style g be used?
low_asterisk:=false;     % should the asterisk be centered at the axis?
math_fitting:=false;     % should math-mode spacing be used?

generate roman           % switch to the driver file

```

Computer Modern

Her er parametrene for standard kursiv
(`cmti10`):

```
% Computer Modern Text Italic 10 point
if unknown cmbase: input cmbase fi

font_identifier:="CMTI"; font_size 10pt#;

u#: =18.4/36pt#;      % unit width
width_adj#: =0pt#;   % width adjustment for certain characters
serif_fit#: =0pt#;   % extra sidebar near lowercase serifs
cap_serif_fit#: =5/36pt#; % extra sidebar near uppercase serifs
letter_fit#: =9.2/36pt#; % extra space added to all sidebars

body_height#: =270/36pt#; % height of tallest characters
asc_height#: =250/36pt#; % height of lowercase ascenders
cap_height#: =246/36pt#; % height of caps
fig_height#: =232/36pt#; % height of numerals
x_height#: =155/36pt#; % height of lowercase without ascenders
math_axis#: =90/36pt#; % axis of symmetry for math symbols
bar_height#: =87/36pt#; % height of crossbar in lowercase e
comma_depth#: =70/36pt#; % depth of comma below baseline
desc_depth#: =70/36pt#; % depth of lowercase descenders

crisp#: =8/36pt#; % diameter of serif corners
tiny#: =8/36pt#; % diameter of rounded corners
fine#: =7/36pt#; % diameter of sharply rounded corners
thin_join#: =7/36pt#; % width of extrafine details
hair#: =11/36pt#; % lowercase hairline breadth
stem#: =23/36pt#; % lowercase stem breadth
curve#: =28/36pt#; % lowercase curve breadth
ess#: =27/36pt#; % breadth in middle of lowercase s
flare#: =29/36pt#; % diameter of bulbs or breadth of terminals
dot_size#: =38/36pt#; % diameter of dots
cap_hair#: =11/36pt#; % uppercase hairline breadth
cap_stem#: =29/36pt#; % uppercase stem breadth
cap_curve#: =34/36pt#; % uppercase curve breadth
cap_ess#: =31/36pt#; % breadth in middle of uppercase s
rule_thickness#: =.4pt#; % thickness of lines in math symbols
```

```

dish#:=1/36pt#;      % amount erased at top or bottom of serifs
bracket#:=20/36pt#;  % vertical distance from serif base to tangent
jut#:=30/36pt#;     % protrusion of lowercase serifs
cap_jut#:=34/36pt#;  % protrusion of uppercase serifs
beak_jut#:=9/36pt#;  % horizontal protrusion of beak serifs
beak#:=70/36pt#;    % vertical protrusion of beak serifs
vair#:=8/36pt#;     % vertical diameter of hairlines
notch_cut#:=10pt#;   % maximum breadth above or below notches
bar#:=11/36pt#;     % lowercase bar thickness
slab#:=11/36pt#;    % serif and arm thickness
cap_bar#:=11/36pt#;  % uppercase bar thickness
cap_band#:=11/36pt#; % uppercase thickness above/below lobes
cap_notch_cut#:=10pt#; % max breadth above/below uppercase notches
serif_drop#:=4/36pt#; % vertical drop of sloped serifs
stem_corr#:=1/36pt#; % for small refinements of stem breadth
vair_corr#:=1/36pt#; % for small refinements of hairline height
apex_corr#:=0pt#;   % extra width at diagonal junctions

o#:=8/36pt#;        % amount of overshoot for curves
apex_o#:=8/36pt#;   % amount of overshoot for diagonal junctions

slant:=.25;         % tilt ratio  $(\Delta x / \Delta y)$ 
fudge:=1;           % factor applied to weights of heavy characters
math_spread:=0;     % extra openness of math symbols
superness:=1/sqrt2; % parameter for superellipses
superpull:=1/6;     % extra openness inside bowls
beak_darkness:=11/30; % fraction of triangle inside beak serifs
ligs:=2;            % level of ligatures to be included

square_dots:=false; % should dots be square?
hefty:=false;       % should we try hard not to be overweight?
serifs:=true;       % should serifs and bulbs be attached?
monospace:=false;   % should all characters have the same width?
variant_g:=true;    % should an italic-style g be used?
low_asterisk:=false; % should the asterisk be centered at the axis?
math_fitting:=false; % should math-mode spacing be used?

generate textit     % switch to the driver file

```

Så hvis man har lyst, kan man meget let lave en personlig udgave af Computer Modern:

```
% TMW Computer Modern
if unknown cmbase: input cmbase fi

font_identifier:="CTMW"; font_size 10pt#;

u#: =20/36pt#;           % unit width
width_adj#: =0pt#;      % width adjustment for certain characters
serif_fit#: =1/36pt#;   % extra sidebar near lowercase serifs
cap_serif_fit#: =3/36pt#; % extra sidebar near uppercase serifs
letter_fit#: =0pt#;     % extra space added to all sidebars

body_height#: =270/36pt#; % height of tallest characters
asc_height#: =250/36pt#; % height of lowercase ascenders
cap_height#: =246/36pt#; % height of caps
fig_height#: =232/36pt#; % height of numerals
x_height#: =120/36pt#;   % height of lowercase without ascenders
math_axis#: =90/36pt#;  % axis of symmetry for math symbols
bar_height#: =92/36pt#; % height of crossbar in lowercase e
comma_depth#: =90/36pt#; % depth of comma below baseline
desc_depth#: =90/36pt#; % depth of lowercase descenders

crisp#: =0pt#;          % diameter of serif corners
tiny#: =11/36pt#;      % diameter of rounded corners
fine#: =6/36pt#;       % diameter of sharply rounded corners
thin_join#: =17/36pt#; % width of extrafine details
hair#: =21/36pt#;      % lowercase hairline breadth
stem#: =42/36pt#;      % lowercase stem breadth
curve#: =27/36pt#;     % lowercase curve breadth
ess#: =25/36pt#;       % breadth in middle of lowercase s
flare#: =29/36pt#;     % diameter of bulbs or breadth of terminals
dot_size#: =38/36pt#;  % diameter of dots
cap_hair#: =21/36pt#;  % uppercase hairline breadth
cap_stem#: =27/36pt#;  % uppercase stem breadth
cap_curve#: =28/36pt#; % uppercase curve breadth
cap_ess#: =27/36pt#;   % breadth in middle of uppercase s
rule_thickness#: =.4pt#; % thickness of lines in math symbols

dish#: =0/36pt#;        % amount erased at top or bottom of serifs
bracket#: =5/36pt#;    % vertical distance from serif base to tangent
jut#: =30/36pt#;       % protrusion of lowercase serifs
```

```

cap_jut#:=32/36pt#;           % protrusion of uppercase serifs
beak_jut#:=10/36pt#;         % horizontal protrusion of beak serifs
beak#:=70/36pt#;             % vertical protrusion of beak serifs
vair#:=21/36pt#;             % vertical diameter of hairlines
notch_cut#:=30/36pt#;        % maximum breadth above or below notches
bar#:=21/36pt#;              % lowercase bar thickness
slab#:=21/36pt#;             % serif and arm thickness
cap_bar#:=21/36pt#;          % uppercase bar thickness
cap_band#:=21/36pt#;         % uppercase thickness above/below lobes
cap_notch_cut#:=1pt#;        % max breadth above/below uppercase notches
serif_drop#:=5/36pt#;        % vertical drop of sloped serifs
stem_corr#:=1/36pt#;         % for small refinements of stem breadth
vair_corr#:=1/36pt#;         % for small refinements of hairline height
apex_corr#:=0pt#;            % extra width at diagonal junctions

o#:=4/36pt#;                 % amount of overshoot for curves
apex_o#:=3/36pt#;            % amount of overshoot for diagonal junctions

slant:=0.1;                   % tilt ratio  $(\Delta x / \Delta y)$ 
fudge:=.95;                   % factor applied to weights of heavy characters
math_spread:=0;               % extra openness of math symbols
superness:=8/11;              % parameter for superellipses
superpull:=1/15;              % extra openness inside bowls
beak_darkness:=4/30;          % fraction of triangle inside beak serifs
ligs:=2;                       % level of ligatures to be included

square_dots:=true;            % should dots be square?
hefty:=false;                  % should we try hard not to be overweight?
serifs:=true;                  % should serifs and bulbs be attached?
monospace:=false;             % should all characters have the same width?
variant_g:=true;              % should an italic-style g be used?
low_asterisk:=false;          % should the asterisk be centered at the axis?
math_fitting:=false;          % should math-mode spacing be used?

generate roman                 % switch to the driver file

```