

# fonts

## The euro symbol

### abstract

When Patrick Gundlach posted a nice METAPOST version of the euro symbol to the ConTeXt discussion list, he added the comment “The official construction is ambiguous: how thick are the horizontal bars? How much do they stick out to the left? Is this thing a circle or what? Are the angles on the left side of the bars the same as the one on the right side? ...” The alternative below is probably not as official as his, but permits a finetuning. You are warned: whatever you try, the euro *is* and *will remain* an ugly symbol.

We use a couple of global variables to control the euro shape within reasonable bounds. Next we define two circles. Next we define a vertical line that we use in a couple of cut and past operations. Watch how the top left point of the outer circle determines the slant of the line that we use to slice the vertical bars.

```
boolean trace_euro ; trace_euro := false ;

vardef euro_symbol = image ( % begin_of_euro

if unknown euro_radius   : euro_radius   := 2cm           ; fi ;
if unknown euro_width    : euro_width    := 3euro_radius/16 ; fi ;
if unknown euro_r_offset : euro_r_offset := euro_width     ; fi ;
if unknown euro_l_offset : euro_l_offset := euro_radius/32 ; fi ;
if unknown euro_l_shift  : euro_l_shift  := euro_r_offset   ; fi ;
if unknown euro_v_delta  : euro_v_delta  := euro_width/4    ; fi ;

save
  outer_circle, inner_circle, hor_bar,
  right_line, right_slant, top_slant, bot_slant,
  euro_circle, euro_topbar, euro_botbar ;

path
  outer_circle, inner_circle, hor_bar,
  right_line, right_slant, top_slant, bot_slant,
  euro_circle, euro_topbar, euro_botbar ;

outer_circle := fullcircle scaled euro_radius ;
inner_circle := fullcircle scaled (euro_radius-euro_width) ;

if trace_euro : for i = outer_circle, inner_circle :
  draw i withpen pencircle scaled 1pt withcolor .5white ;
endfor ; fi ;

right_line :=
  (lrcorner outer_circle -- urcorner outer_circle)
  shifted (-euro_r_offset,0) ;

outer_circle := outer_circle cutbefore right_line ;
```

```

right_slant :=
  point 0 of outer_circle
  -- origin shifted (0,ypart lrcorner outer_circle) ;

euro_circle := buildcycle(outer_circle, right_line,
  reverse inner_circle, reverse right_slant) ;

hor_bar := (-euro_radius,0) -- (euro_radius,0) ;

top_slant :=
  right_slant shifted (-euro_radius+euro_r_offset-euro_l_offset,0) ;

bot_slant :=
  top_slant shifted (0,-euro_l_shift) ;

if trace_euro : for i = right_line, right_slant, top_slant, bot_slant :
  draw i withpen pencircle scaled 1pt withcolor .5white ;
endfor ; fi ;

euro_topbar := buildcycle
  (top_slant, hor_bar shifted (0, euro_v_delta),
  right_slant, hor_bar shifted (0, euro_v_delta+euro_width/2)) ;

euro_botbar := buildcycle
  (bot_slant, hor_bar shifted (0,-euro_v_delta),
  right_slant, hor_bar shifted (0,-euro_v_delta-euro_width/2)) ;

for i = euro_circle, euro_topbar, euro_botbar :
  draw i withpen pencircle scaled 0 ;
endfor ;
for i = euro_circle, euro_topbar, euro_botbar :
  fill i withpen pencircle scaled 0 ;
endfor ;

if trace_euro :
  drawpoints euro_circle withcolor red ;
  drawpoints euro_topbar withcolor green ;
  drawpoints euro_botbar withcolor blue ;
fi ;

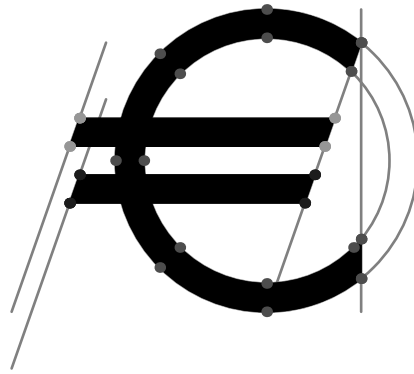
) enddef ; % end_of_euro

```

We only set a parameter when it is not yet set. This has the advantage that we don't have to set them when we change one. This way of manipulating paths (cutting and building) does not always work well because of rounding errors, but here it does work.

```
euro_radius := 4cm ; trace_euro := true ; draw euro_symbol ;
```

For educational purposes, we have added as a bit of tracing. When enables, the euro shows up as:



Of course it would be best to define the euro as one shape, but we won't go through that process right now. By packaging the combined paths in an image, we can conveniently color the euro symbol

```
draw euro_symbol withcolor .625red ;
```



You may wonder why we both draw and fill the euro, using a pen with zero width. We've done this in order to demonstrate the `redraw` and `refill` macros.

```
redraw currentpicture withpen pencircle scaled 4pt withcolor .625yellow ;
refill currentpicture withcolor .625white ;
setbounds currentpicture to boundingbox currentpicture enlarged 2pt ;
```

