

Writing in multiple directions in Omega*

John Plaice[†] and Yannis Haralambous[‡]

Preliminary — 25 February 2001

Abstract

We introduce the basics of the writing direction primitives used in the Omega Typesetting and Document Processing System. A writing direction has three components, designating the top of the page, the beginning of the line and the orientation of the characters. We give the details of each of these directions and explain how the directions of text, boxes and pages can be controlled.

1 Introduction

One of the key assumptions in most text processing software is that writing takes place horizontally, with text flowing left-to-right, with successive lines flowing top-to-bottom. This form of writing is used by much of the world's population, but by no means by all.

Arabic, Hebrew and several other scripts originating in the Middle East are written horizontally, from right-to-left. Traditional writing and printing in East Asia, including Japan, is done vertically, with the first line of the page on the right hand side. Uighur and Mongolian writing is also vertical, but the first line of the page is on the left.

To complicate matters, it is possible for different scripts, which have different “natural” directions, to be intermixed on the same page, thereby creating some situations where a script may be typeset in a rotated manner, in order for it to fit in a broader context.

This paper explains how these problems are solved by Omega in their full generality. We begin by summarizing the writing directions, as defined in Omega. We then look in detail at how each of the directions works, using modified versions of the `cmr10` font. We then explain the Omega parameters that allow one to control the mixing of writing directions.

2 The writing directions

Omega assumes that a writing direction can be designated by three characters, where each is one of Top, Bottom, Left, and Right. These characters absolutely designate one of the edges of the physical page. Then a writing direction must designate:

Primary part. The “top” of each page.

Secondary part. The “left” of each page.

Tertiary part. The “top” of each character.

The secondary direction must be orthogonal to the primary direction. The tertiary direction can take all four values. Hence there are 32 possible directions. Here are the most common ones:

*Presented to the *Fourth International Symposium on Multilingual Computing*, March 1-3, Tokyo, Japan.

[†]John Plaice, School of Computer Science and Engineering, The University of New South Wales, UNSW SYDNEY NSW 2052, Australia. plaice@cse.unsw.edu.au

[‡]Yannis Haralambous, Atelier Fluxus-Virus, 187 rue Nationale, F-59800 Lille, France. yannis@fluxus-virus.com

TLT — Left-right scripts, horizontal CJK.
TRT — Right-left scripts.
RTT — Vertical CJK, upright left-right scripts in vertical CJK.
RTL — Mongolian in vertical CJK.
RTR — Rotated left-right scripts in vertical CJK.
LTL — Mongolian.
LTR — Rotated left-right scripts in Mongolian.
LTt — Vertical CJK in Mongolian.

3 The basic directions

From a given glyph, we can, using reflection and rotation, derive 8 possible glyphs. For each of these glyphs, there are two possible metrics. Therefore, using .tfm files, we get 16 fonts derived from a fixed set of glyphs.

When the glyphs are rotated by 90 degrees, then some of the punctuation glyphs (parentheses, brackets and dashes) are not rotated.

Below, we have taken the `cmr10` font and produced the 16 possible derived fonts. We give the crucial METAFONT code that expresses the rotation, reflection and translation of glyphs, including, as needed the changes for punctuation glyphs.

We give two example paragraphs for each font variation, with ‘down’ and ‘up’ writing.

3.1 Directions TLT and BLT

Omega is
currently
(typesetting)
in the TLT
direction.

direction.
in the BLT
(typesetting)
currently
Omega is

```
font_identifier:="CMR";
```

3.2 Directions TLL and BLL

Ωϖεβσζ .ις
 ςβηηεδρ ιλ
 (ρλρσσερ ριρσζ)
 .ι ρ ρε εηη
 ριρσζ ριρσζ

direction.
in the BLB
(typesetting)
currently
Omega is

```
font_identifier:="D_LL_CMR";

def doparencorrect =
  currentpicture:=currentpicture shifted (0,.5(d-h))
enddef;

def docorrect =
  currentpicture:=currentpicture rotated 90;
  currentpicture:=currentpicture shifted (h,-.5w)
enddef;
```

3.3 Directions TLB and BLB

Omega is
currently
(typesetting)
in the TLB
direction.

direction.
in the BLB
(typesetting)
currently
Omega is

```
font_identifier:="D_LB_CMR";

def docorrect =
  currentpicture:=currentpicture reflectedabout ((0,0),(1,0));
enddef;
```

3.4 Directions TLR and BLR

Omega is
currently
(typesetting)
in the TLR
direction.

direction.
in the BLB
(typesetting)
currently
Omega is

```
font_identifier:="D_LR_CMR";
```

```
def doparencorrect =
  currentpicture:=currentpicture shifted (0,.5(d-h))
enddef;
```

```
def docorrect =
  currentpicture:=currentpicture reflectedabout ((0,0),(0,1));
  currentpicture:=currentpicture rotated 90;
  currentpicture:=currentpicture shifted (h,.5w);
enddef;
```

3.5 Directions TRT and BRT

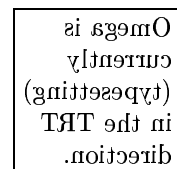


Diagram showing the transformation sequence for TRT: direction, in the TRT, (typsetting), currently, Omega is.

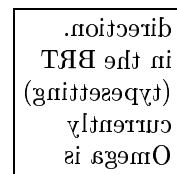


Diagram showing the transformation sequence for BRT: Omega is, currently, (typsetting), in the BRT, direction.

```
font_identifier:="D_RT_CMR";
```

```
def docorrect =
  currentpicture:=currentpicture reflectedabout ((0,0),(0,1));
enddef;
```

3.6 Directions TRR and BRR

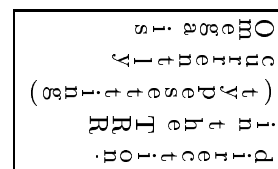


Diagram showing the transformation sequence for TRR: Omega is, currently, (typsetting), in the TRR, direction.

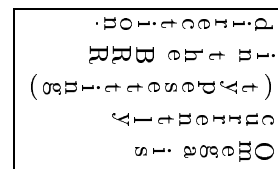


Diagram showing the transformation sequence for BRR: Omega is, currently, (typsetting), in the BRR, direction.

```
font_identifier:="D_RR_CMR";
```

```
def doparencorrect =
  currentpicture:=currentpicture shifted (0,.5(d-h));
  currentpicture:=currentpicture reflectedabout ((0,0),(0,1))
enddef;
```

```

def docorrect =
  currentpicture:=currentpicture rotated -90;
  currentpicture:=currentpicture shifted (-h,.5w);
enddef;

```

3.7 Directions TRB and BRB

```
font_identifier:="D_RB_CMR";
```

```

def docorrect =
  currentpicture:=currentpicture rotated 180;
enddef;

```

3.8 Directions TRL and BRL

```
font_identifier:="D_RL_CMR";
```

```

def doparencorrect =
  currentpicture:=currentpicture shifted (0,.5(d-h));
  currentpicture:=currentpicture reflectedabout ((0,0),(0,1));
enddef;

```

```

def docorrect =
  currentpicture:=currentpicture reflectedabout ((0,0),(0,1));
  currentpicture:=currentpicture rotated -90;
  currentpicture:=currentpicture shifted (-h,-.5w);
enddef;

```

3.9 Directions LTR and RTR

direction.
in the LTR
(typesetting)
currently
Omega is

Omega is
currently
(typesetting)
in the RTR
direction.

```
font_identifier:="D_TR_CMR";
```

```
def docorrect =  
  currentpicture:=currentpicture rotated -90;  
enddef;
```

3.10 Directions LTT and RTT

direction.
in the LTT
(typesetting)
currently
Omega is

Omega is
currently
(typesetting)
in the RTT
direction.

```
font_identifier:="D_TT_CMR";
```

```
def doparencorrect =  
  currentpicture:=currentpicture rotated -90;  
  currentpicture:=currentpicture shifted(.5(d-h),0);  
enddef;
```

```
def docorrect =  
  currentpicture:=currentpicture shifted (-.5w,-h);  
enddef;
```

3.11 Directions LTL and RTL

direction.
in the LTL
(flipped)
currently
Omegas is

Omegas is
currently
(flipped)
in the RTL
direction.

```
font_identifier:="D_TL_CMR";
```

```
def docorrect =
  currentpicture:=currentpicture reflectedabout ((0,0),(1,0));
  currentpicture:=currentpicture rotated -90;
enddef;
```

3.12 Directions LTB and RTB

direction.
in the LTB
(flipped)
currently
Omegas is

Omegas is
currently
(flipped)
in the RTB
direction.

```
font_identifier:="D_TB_CMR";
```

```
def doparencorrect =
  currentpicture:=currentpicture rotated -90;
  currentpicture:=currentpicture shifted(.5(d-h),0);
enddef;
```

```
def docorrect =
  currentpicture:=currentpicture reflectedabout ((0,0),(0,1));
  currentpicture:=currentpicture shifted (.5w,-h);
enddef;
```

3.13 Directions LBL and RBL

Omega is
currently
(typesetting)
in the LBL
direction.

direction.
in the RBL
(typesetting)
currently
Omega is

```
font_identifier:="D_BL_CMR";
```

```
def docorrect =  
  currentpicture:=currentpicture rotated 90;  
enddef;
```

3.14 Directions LBB and RBB

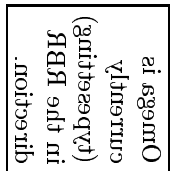
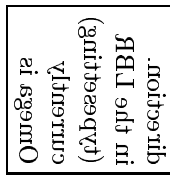
Omega is
currently
(typesetting)
in the LBB
direction.

direction.
in the RBB
(typesetting)
currently
Omega is

```
font_identifier:="D_BB_CMR";
```

```
def doparencorrect =  
  currentpicture:=currentpicture rotated 90;  
  currentpicture:=currentpicture shifted (.5(h-d),0);  
enddef;  
  
def docorrect =  
  currentpicture:=currentpicture rotated 180;  
  currentpicture:=currentpicture shifted (.5w,h);  
enddef;
```

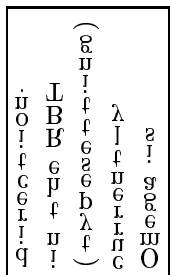
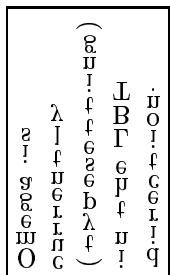
3.15 Directions LBR and RBR



```
font_identifier:="D_BR_CMR";
```

```
def docorrect =
  currentpicture:=currentpicture reflectedabout ((0,0),(0,1));
  currentpicture:=currentpicture rotated -90;
enddef;
```

3.16 Directions LBT and RBT



```
font_identifier:="D_BT_CMR";
```

```
def doparencorrect =
  currentpicture:=currentpicture rotated 90;
  currentpicture:=currentpicture shifted(.5(h-d),0);
enddef;

def docorrect =
  currentpicture:=currentpicture reflectedabout ((0,0),(0,1));
  currentpicture:=currentpicture rotated 180;
  currentpicture:=currentpicture shifted (-.5w,h);
enddef;
```

4 Controlling writing directions

Omega now includes a number of *direction parameters*. They are

- `\pagedir dir`. The direction of the current page, as used for headers, footers, footnotes, and such things.
- `\bodydir dir`. The direction of the main part of the page, i.e. the real text.
- `\pardir dir`. The direction of the current paragraph.
- `\textdir dir`. The direction of the text within the current paragraph.
- `\mathdir dir`. The direction of mathematics when it appears.
- `\boxdir n dir`. The direction of box *n* when it appears.

These act exactly as integer parameters, where integers are replaced by three letter codes representing directions. Hence if you type `\the\boxdir45`, the direction of box 45 will be placed on the current stream.

The default for all of these values is TLT, the direction for left-to-right scripts. Changes to these values all respect grouping, so it is possible to have direction changes that span paragraphs, if desired. If the page direction is changed, then the current vertical list is cleared until it is empty, to avoid any weird situations.

In addition to the above parameter, there is also `\nextfakemath`, which forces the next entrance to math mode to use `\the\textdir` instead of `\the\mathdir`. This is needed because many table environments use math mode.

This paragraph is in TRT but the text is in TLT, and now we have changed it so that
it receives insertion with a TRT insertion point. We are
back to the TLT text.

Finally, `aligns` and `box` definitions can include a `dir` *dir* argument. For example, the box ⁹

was produced with the text

`\hbox dir RTT{\fontRTT Hello, there}`

5 Conclusion

Defining these primitives is absolutely necessary for being able to do real multilingual documents. However, there is still a lot of work to be done at the macro level in order to create robust macro packages that can take advantage of these new features. These macro packages may require further parametrization, which will become obvious as the primitives start being used.