



Collabora Productivity

Bottom to top, left to right writing direction in Writer

By Miklos Vajna

Software Engineer at Collabora Productivity

2019-09-11

About Miklos

From Hungary

- More details: <https://vmiklos.hu/>

Google Summer of Code 2010 / 2011

- Rewrite of the Writer RTF import/export

Then a full-time LibreOffice developer for SUSE

Now a contractor at Collabora

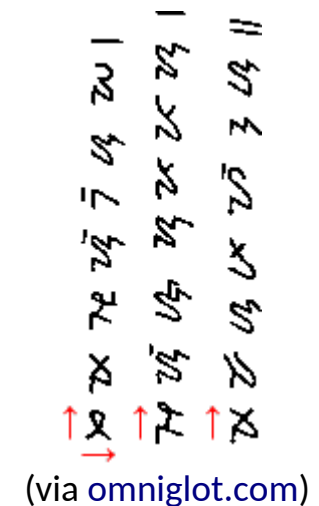
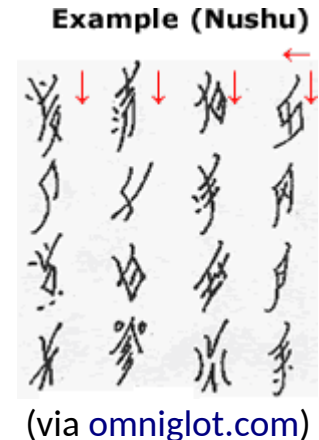
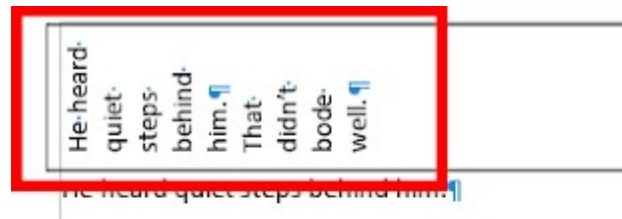


**btLr writing direction
in Writer**

Motivation: rotated text in table cells

Rotated text: rotate to left or rotate to right?

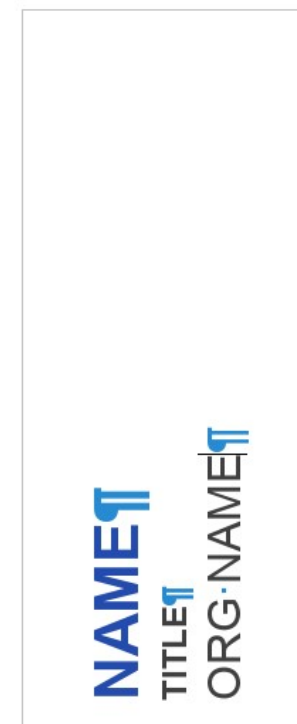
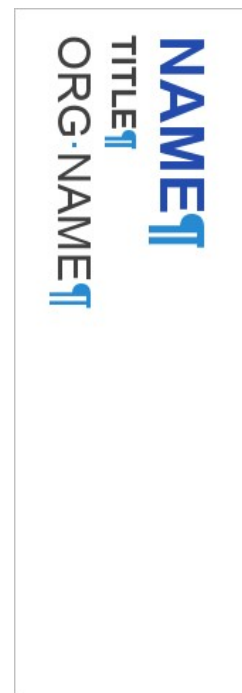
- Rotate to right is top to bottom, right to left
 - Also used for e.g. Japanese text
 - Has good support already
- Rotate to left is bottom to top, left to right
 - Also used for languages in Indonesia and Philippines
 - Poor support in the past, now works:



Rotated text in text frames

Use-case: place cards

- ↔ btLr table cells are typically headers for rows in Latin text
- Write the name of the person in two text frames on the left and right of the page
- Rotate to the left and right, print, fold it in the middle vertically
- Complex content: only possible with Text Frames, but rotate to left was broken
- Now works:



Cursor travel

Up, left, down, right arrows should work differently

- Physical up: document model right
- Physical left: document model up
- Physical down: document model left
- Physical right: document model down
- Also rotate the cursor caret's line
- Now works:

He heard quiet steps behind him. That didn't bode well.	
He heard quiet steps behind him. That didn't bode well. Who could be following him this late at night and in this deadbeat part of town? And at this particular moment, just after he pulled off the	

Cursor selection

Cursor is primarily a document model concept

- A list of start/end pairs: paragraph and character index
- This is then transformed to a list of rectangles
- If the text is rotated, then expectation:
 - Selection overlay is rotated, too
- Now works:



Filters: ODF

ODF proposal pending to support this

- Extension namespace for now
- Collabora Office 6.2 supports both cells and frames
- LibreOffice 6.3 supports cells + reads frames
- LibreOffice 6.4 supports creating frames

```
<style:style style:name="Table1.A1" style:family="table-cell">
  <style:table-cell-properties loext:writing-mode="bt-lr"/>
</style:style>
<style:style style:name="Table1.B1" style:family="table-cell">
  <style:table-cell-properties/>
</style:style>
<style:style style:name="Table1.C1" style:family="table-cell">
  <style:table-cell-properties style:writing-mode="tb-rl"/>
</style:style>
```



Filters: DOCX, DOC, and RTF

Word supported this for a long time already

- DOCX import/export cares about shape text, not text frames
 - Underlying code is shared, though
- Export of Writer textframes to DOCX is handled separately
- Similar story for binary DOC and RTF

```
<w:tr>
  <w:tc>
    <w:tcPr>
      <w:textDirection w:val="btLr"/>
    </w:tcPr>
  </w:tc>
  <w:tc>
</w:tc>
  <w:tc>
    <w:tcPr>
      <w:textDirection w:val="tbRL"/>
    </w:tcPr>
  </w:tc>
</w:tr>
```



**How is this
implemented?**

Implementation: document model

SvxFrameDirection in editeng

- The direction / writing mode enumeration used even for cells
- New Vertical_LR_BT, then use this everywhere
- This replaces the old hack:
 - Assume that all text is a single paragraph
 - Rotate content at a character level, where this direction already supported
- Reuses existing VCL API, that one only cares about the text angle



Implementation: public / stable UNO API

No compatibility concern here

- `text::WritingMode2` is already a constant group
- New `text::WritingMode2::BT_LR`
- Import/export is easy now:
 - Instead of iterating all paragraphs and text portions, do rotation at a table cell / text frame level
 - Just one more case, similar to `tbRI`



(via [richcampbell](#))

Implementation: layout

Ignoring RTL, we had Hori, Vert and VertLR directions

- Now we have a new VertLRBT one
- The core of it is a SwRectFnCollection instance: aVerticalLeftToRightBottomToTop
 - Allows client code to be generic and get e.g. width, which will be “right – left” for Hori, but “bottom – top” for VertLRBT
- Also: a SwFrame::mbVertLRBT, and SwFrame::isVertLRBT()
- For code that does not use SwRectFnCollection:
SwTextFrame::SwitchHorizontalToVertical()
 - int, Point, SwRect versions * converting back and forth * “is swapped or not”
→ 12 cases for 1 writing direction
 - Need to support VertLRBT here, too
- This was obviously the most challenging part of the work
 - I tried this once already 6 years ago, and this where I gave up :-)

Implementation: filters

ODF filter was tricky, due to compatibility

- Can't just write `style:writing-mode="bt-lr"`
 - But then we assumed that a group of constants always map to the same XML attribute
 - Custom code for both table cells and frames
- Word formats:
 - DOCX is a bit special: there text frames are actually shapes with textboxes, so needed to bridge the shape's text angle with the `btLr` support of the underlying text frame
 - DOC and RTF was straightforward, got rid of `WW8TabDesc::Start/EndMiserableHackForUnsupportedDirection()` :-)

Implementation: tests

Tests are added at multiple levels

- Filter tests: ODF, DOCX, DOC, RTF
- CppunitTest_sw_layoutwriter: layout-level testing
 - Just tests a hardcoded position
 - Manual testing shows that our PDF out and Word's PDF output has no diff
- CppunitTest_sw_uiwriter: typing → correct invalidation
- UITest_table: covers table cell / btlr code to create such a table cell

Implementation: user interface

Table cells

- Handled in SwTextFlowPage
- Existing “Vertical” is now “Vertical (top to bottom)”
- Then a new “Vertical (bottom to top)”
- Text frames are similar in SwFrameAddPage:
 - Existing “Right-to-left (vertical)” unchanged
 - New “Bottom-to-top, left-to-right (vertical)” option there

Implementation: specification

We now try to track the ODF extensions a bit more carefully

- schema/ has the proposed new “bt-lr” writing direction included
- <https://issues.oasis-open.org/browse/OFFICE-4030> tracks the proposal status
- <http://www.w3.org/TR/2006/REC-xsl11-20061205/> already contains “bt-lr” and that’s what ODF reuses, so should be no problem
- Except that ODF currently reuses the old <http://www.w3.org/TR/2001/REC-xsl-20011015/> which doesn’t have it

Thanks

Collabora is an open source consulting and product company

- What we do and share with the community has to be paid by someone

There is an exception to every rule

- This work was done as a Collabora Hackweek project

Summary

Writer now has much better support for btLr text

- Headers to table rows
- Rotated text frame / shape text
- Available in Collabora Office 6.2 snapshots
- Available in LibreOffice 6.3 (table cells) and 6.4 (text frames)

Thanks for listening! :-)

- Slides: <https://vmiklos.hu/odp>