Layout-true Representation of OOXML Documents: Use Case 3 Report

List formatting in .docx

Assignment

Lists (numbered or otherwise) formatted in Microsoft Word and saved in OOXML format have to be imported as a true representation into LibreOffice/OpenOffice, and subsequently exported back to OOXML.

- The dimensions (e.g. indents) match the original document in size and appearance.
- The bullet point symbols match the original document.
- The numbering convention matches the original document.
- The nesting matches the original document.
- The document structure is not changed when importing from the subsequent OOXML format export.

Notes

Problems in displaying bullet points are often related to unavailable fonts. A pragmatic approach that sees the bullet point symbols hardcoded in LibreOffice/OpenOffice is conceivable.

The document structure must not be changed so as to prevent unwanted problems with any automated content that it supports (e.g. tables of contents or similar) as well as VBA macros.

Test Documents

See files in the relevant folder on Filespots.

Expectations

From the technical point of view, it is unrealistic to promise that any random document will result in a true representation, ie. accurate positioning of all pixels at a resolution of 70 dpi. We can commit to best-effort handling of the documents that will be provided as test examples, and in cases where it is technically impossible to achieve true representation, we will explain the barriers, and implement a solution that is as close to the true representation as possible.

Work Description

We focused on fixing the example documents that were provided as part of the tender in the https://freiburg.filespots.com file repository. It contained the following three problematic documents:

1. The numbering was wrong in the case the paragraph had its own <w:numPr>. This was fixed with: http://cgit.freedesktop.org/libreoffice/core/commit/?id=e7ab4bb6b0e83f01148ffff41e8c5eaa0c5ba0a4
   An additional fix was necessary for the ODF roundtrip: http://cgit.freedesktop.org/libreoffice/core/commit/?id=fc508908f55cc1fe5a22adcba710cebb75fc979c
   There was a remaining issue, and that was that the indenting was too far away. After some investigation, it turned out to be caused by a missing font, in this case Cambria, which is licensed and distributed with MS Office. As the font related work was covered by the Use Case 5, we did not proceed with this until later.

2. The second document had the same problem as the 1st one, that means the commits mentioned above fixed even this case.

3. The default tab stop was incorrectly written to the default style during roundtrip. This was fixed: http://cgit.freedesktop.org/libreoffice/core/commit/?id=15af925c254f27046427de70a59011e2ac3d6bdb

The above fixes were done in about 3 days, and no other test documents were provided. As a consequence, we agreed with OSBA to spend the remaining time (about 7 days) on preliminary work on the Use Case 5: Embedding Fonts in OOXML and ODF.

We focused on reading (and writing again) the font information embedded in .docx. As a result, basic .docx embedded font reading and writing is supported, sufficient for practical use. If a .docx document with embedded fonts is read, fonts are also embedded when saving the document.

Limitations:

- only .docx support
- no UI to enable font embedding for saving (document needs to be first saved in MSO with the option enabled)
- no special handling of subsetted fonts (they are read, used and saved, but it is ignored that they are subsetted). The usage of subsetted fonts seems questionable - non-subsetted fonts make the documents noticeably larger, but subsetted fonts e.g. may not contain necessary glyphs for further document editing.

The code is here:

http://cgit.freedesktop.org/libreoffice/core/commit/?id=c1c8adca05b561afbbf3346b73225d80f2b82ee4
http://cgit.freedesktop.org/libreoffice/core/commit/?id=b7e56788135c1c6179cbc5387e41a66a85a7460b
http://cgit.freedesktop.org/libreoffice/core/commit/?id=9b14fa8f64d84866777e35acfe369503da188c7a
http://cgit.freedesktop.org/libreoffice/core/commit/?id=7a045f48bad2177538c43f76019c1caecdd5baf7
http://cgit.freedesktop.org/libreoffice/core/commit/?id=cf6d2e2f8319fb4a2b15b9a805699312fe7305f9
http://cgit.freedesktop.org/libreoffice/core/commit/?id=11f7d6aca36b25fb0b225cd0c641cd4f09338672

http://cgit.freedesktop.org/libreoffice/core/commit/?id=47919e05f68a6871031b92ad028e4345a51bf5c9

http://cgit.freedesktop.org/libreoffice/core/commit/?id=fc169270eaeb8156d40740cd088cd8ed958ea99c

Conclusion

We have fixed the documents that were provided to us. Because we fixed that earlier, and one of the document showed issue that is covered by Use Case 5, font embedding, we agreed with OSBA to provide limited work on the font embedding too. We were successful, and implemented the reading and writing of embedded fonts in .docx files. It is of course by no means a complete solution of Use Case 5 – in order to be finished, ODT roundtrip, and user interface has to be implemented. Additionally, smaller fixes in internal structures are necessary too.

All the code we delivered is committed to LibreOffice repository.