Specification of

Major Feature Improvements for
LibreOffice/Apache OpenOffice

24 March 2014

Open Source Business Alliance (OSBA) Working Group
"Office Interoperability"
Introduction

This specification document forms the basis on which software companies can develop tenders for the technical implementation of the requirements described below. The specification itself is based on the results of a workshop held in Stuttgart between on 29 and 30 October 2013, in which public institutions and LibreOffice/Apache OpenOffice developers discussed the requirements that shall be implemented.

Objective

The following requirements aim to improve major features within LibreOffice/Apache OpenOffice. Functionalities such as mail merge in Writer or diagrams in Calc are often applied by professional users for high volumes of data. Therefore these important features need to be available in a highly user-friendly way and work reliable in every situation.

Another important feature of today's office suites is change tracking within documents. LibreOffice and Apache OpenOffice offer change tracking in ODF files. However, Microsoft Office has not implemented change tracking for ODF documents since Microsoft says the current change tracking specification with the ODF standard is defined insufficiently. Therefore Use Case 6 of these requirements does cover the exact specification of change tracking within the ODF standard in order to enable Microsoft to implement it in future versions of Microsoft Office.

Institutional Crowdfunding

For this development project several public institutions and “office user” companies are willing to fund bidder companies to implement the requirements. The project therefore follows the workflow of an institutional crowdfunding initiative (see figure below) leading to jointly funded open source software development. With the publication of this specification open source development companies are invited to estimate the effort for the required work and to place a tender for one or more of the Use Cases explained below.

Phase 1: Initialization

a) Mobilize interest of institutional open source software users, find funding for specification
b) Create clear and common understanding of the issues, ask the experts
c) Result: aggregated requirements, clustered as Use Cases within a specification

Continue only if previous phase is completed successfully

Phase 2: Funding

a) Publish specification as Request for Proposal (RfP), invite companies to offer
b) Evaluate and decide for best proposal(s) for each of the Use Cases
c) Result: find funding for each Use Case from institutional open source software users and other sources to implement the specification

Continue only if previous phase is completed successfully

Phase 3: Implementation

a) Define project management, sign contracts, start implementing
b) Do testing among open source software users, finalize development
c) Result: Publish new source code, pass it upstream to the open source project

Institutional open source crowdfunding method
OSBA Working Group Office Interoperability

This project is being managed by the Working Group “Office Interoperability” of the Open Source Business Alliance (OSBA). The Working Group was founded 2011 and is open to all members of the OSBA. This is the link to the official website of the Working Group:

http://www.osb-alliance.de/working-groups/wg-office-interoperability/

The Working Group has appointed Dr. Matthias Stürmer (stuermer@osb-alliance.com) as spokesperson. He acts as the single point of contact for the project's communication and coordination. The Working Group spokesperson publishes the specification, receives tenders from software companies, coordinates the process of appointing contractors, prepares the master agreement, coordinates financing agreements and the signing of contractual agreements, monitors implementation progress and coordinates the software development process.

However, the spokesperson does not have the authority to sign any development contract in the name of the Working Group. Orders are signed directly by the funding institutions with the contractors. Accordingly, the implementation work is divided into Use Cases which are independent working packages.

Schedule (subject to change)

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>24 March 2014</td>
<td>Publication of this specification and test documents</td>
</tr>
<tr>
<td>30 April 2014</td>
<td>Deadline for tenders from software companies</td>
</tr>
<tr>
<td>31 May 2014</td>
<td>End of selection process, desired contractor is appointed</td>
</tr>
<tr>
<td>June/July 2014</td>
<td>Funding of the required cost is conducted among professional users</td>
</tr>
<tr>
<td>August 2014</td>
<td>Contracts between sponsors and implementers are signed, implementation starts</td>
</tr>
<tr>
<td>November 2014</td>
<td>All requirements are implemented, tested and accepted</td>
</tr>
<tr>
<td>End of 2014</td>
<td>Requirements are integrated in LibreOffice and Apache OpenOffice, source code is published under the Apache Software License 2</td>
</tr>
</tbody>
</table>
General Conditions

The general conditions describe under which circumstances this project shall be offered and implemented.

Terminology

- Microsoft Word, Microsoft Excel and Microsoft PowerPoint refer to 2007, 2010, and 2013 versions of the program.
- OOXML is the general term used in this document for the new .docx, .xlsx and .pptx XML-based document formats introduced in Microsoft Office 2007.
- "Valid" is a validation of the XML for ODF 1.0/11 as per the "strict" requirements and for ODF 1.2 as per the "conformant" requirements.

Use Cases and Features

The requirements are split into a separate Use Case. Each Use Case is split into separate Features. An offer may include the implementation of one or more Use Cases. Within the offer the effort required for each of the Use Cases as well as each of the Features must be reported. For each of the Use Cases and the Features a) the programming work, b) the integration work required for LibreOffice, and c) the integration work required for Apache OpenOffice must be indicated in the tender.

Operating Systems

All requirements need to be integrated in the office versions for Linux, Mac, and Windows platforms.

Code Basis

The code implementing the use cases must be delivered to OSBA in two versions: one suitable for LibreOffice integration, one suitable for Apache OpenOffice integration. That means each version of the code changes must be developed, rebased and tested against a commit of the respective project that is not older than 60 days at the time of the delivery.

Binaries of both LibreOffice and Apache OpenOffice containing the changes have to be provided for testing. Each Use Case is completed if the code and binaries are provided to OSBA, and the features match the specification.

License

The contractor must publish the software source code developed for this project under license suitable for integration in the respective projects, ideally dual licensed Apache License Version 2.0, and Mozilla Public License Version 2.0. The contractor may choose to publish under Apache License Version 2.0 only. Respective projects are encouraged to accept such changes, but such acceptance is not a prerequisite for Use Case completion.
Technical Requirements

Each Use Case consists of a detailed description (see below) and possibly some test documents of the functionality requested. The test documents are freely available on the OSB Alliance file sharing website of the Working Group Office Interoperability:

https://files.osb-alliance.com/owncloud/public.php?service=files&t=eff3851d03d99fd225280ce085eb7723

Use Case 1: Improve mail merge in Writer

The mail merge function within Writer needs to be made more user-friendly to handle. The following Features are required to be implemented for this Use Case:

1. **Database connection of a mail merge Writer document should be included in the text file.**

   Description: When a mail merge document is being saved a separate database file is being automatically created (see screenshot below). It links the Writer file to the spreadsheet (ODT, XLS, XLSX etc.) as data source. This additional 2kB ODB file confuses users, they might delete it without knowing to break the connection to the data source. Also this ODB file makes it more difficult to reuse the Writer file with other data sources. The necessary connector as well as all configurations for the mail merge procedure should be stored in the Writer file.

![Database connection screenshot]

2. **A two page document should be previewed by a customized two page document and not by duplicating the document.**

   Description: A document with for example two pages and some merge fields is currently displayed by the mail merge wizard the same number of times there are entries in the database. In Microsoft Word however the same document is shown only on two pages. To edit the entries you can click on the arrow key to the next or the previous entry. LibreOffice/Apache OpenOffice should behave the same.

![Two page document preview screenshot]
3. In the printer dialog the number of pages to be printed should correspond to the number of generated pages from the mail merge function.

Description: Currently the number of pages in the print dialog is almost the double of pages because of two sided printing as the screenshot shows below.

![Printer dialog screenshot](image1)

4. Printer default preferences of the operating system need to be applied when using the mail merge function.

Description: Currently default printer preferences are not applied for mail merge printing.

![Printer settings screenshot](image2)

Use Case 2: Improve paragraph handling in Writer

Handling of paragraphs in Writer should be improved. The following Features are required to be implemented for this Use Case:

1. A Text frame should move to the first page in the given document when the text on the first page is deleted.

Description: A document has a “TextFrame” on the second page as the screenshot shows below and as shown in `Testdocument_UC2-1_UpdateTextarea.odt`. Even if there is enough place on the first page, the “TextFrame” stays at the top of the second page. The only workaround consists in saving the document, closing and reopening it. The TextFrame then jumps to the first page.

![Paragraph handling screenshot](image3)
2. Auto text categories should be sorted automatically and alphabetically.

Description: If there is a document with customized categories of Autotextes (see screenshot below), they are not sorted alphabetically. (This bug is solved in OpenOffice 4.0.x but not in the latest version of LibreOffice 4.1.x)

3. Special characters should be supported for Autotext abbreviation

Open the Autotext edition dialog via Edit - Autotext. Currently it is possible to create an Autotext with numbers as a shortcut (see screenshot below).
Description: As in the example above, abbreviations like 10.1 or 10.1.1 should be allowed. Actually, it is possible to create an Autotext with the above abbreviation, but if your type the following: 10.1 + <F3>, the Autotext won't be called

But if one first selects the number 10.1 with the mouse, after pressing the <F3> key the Autotext will be called. It shouldn't be necessary to select the number with the mouse for calling the Autotext.

4. The name and abbreviation field should work as an "auto-completion" field

The name and abbreviation field should work like a completion filter in the Autotext edit dialog (see Use Case 2.4). Typing a character should filter the list of available Autotexts in the list below according to the typed characters.

5. The different parts of an image (caption, image and source) should stick together even if the image is moved.

Description: When an image is moved by drag and drop (see test document Testdocument_UC2-3_DocumentWithImageAndCaption.odt), the image caption and other text is currently not moved as the two screenshots show below.

![An image with caption before moving.](image1.png)

![After moving the image caption stays within the text frame. It should stick with the picture.](image2.png)

6. Paragraph property “Keep with next paragraph” should always work.

Description: In paragraph properties the checkbox “Keep with next paragraph” allows two paragraphs to stick together so there is no page break between them. Sometimes this property does not work. A page break is inserted between the paragraphs even if the option "Keep with next paragraph" is selected (see screenshot below and test document Testdocument_UC2-4_KeepWithNextParagraph.odt). Only if the yellow highlighted text is edited the paragraph jumps on the next page.
Use Case 3: Implement styles in all content elements of Writer

The implement styles should be available in all content elements of Writer. The following Feature is required to be implemented for this Use Case:

1. **Paragraph styles should be available in graphical objects such as shapes.**
   
   Description: Currently styles are not available for shapes and other graphical elements within Writer as the screenshot shows below.
Use Case 4: Add chart styles in Calc

The diagrams in Calc should provide better ways to format several diagrams at once. The following Feature is required to be implemented for this Use Case:

1. It should be possible to do text formatting, change labels, and apply general properties to all diagrams in a spreadsheet at the same time. This could be done by chart styles.

Description: Currently if there are several diagrams in a Calc sheet their text, label and other styles cannot be edited at the same time for all diagrams as the screenshot below and the test document Testdocument_UC4-1_DiagramStyle.ods show.

![Screenshot of diagrams in Calc](image)

Use Case 5: Make more functions available in shared spreadsheets in Calc

Shared spreadsheets should offer more functions than they do now. The following Features are required to be implemented for this Use Case:

1. **Add the following functions to the Edit menu items:**
   a) Changes (including all sub menu items)
   b) Compare Document
   c) Sheet > Move/Copy & Delete

2. **Add the following functions to the Insert menu items:**
   a) Cells Shift Cells Down & Shift Cells Right
   b) Sheet from file
   c) Names

---

d) Comment
e) Picture > From File
f) Movie and Sound
g) Object
h) Chart
i) Floating Frame

3. **Add the following functions to the Format menu items:**
   a) Rename, Tab Color
   b) Merge Cells > Merge and Center, Merge Cells, Split Cells
   c) Print Ranges

4. **Add the following functions to the Tools menu items:**
   a) Protect Document

5. **Add the following functions to the Data menu items:**
   a) Define Range
   b) Sort
c) Subtotals
d) Validity
e) Multiple Operations
f) Consolidate
g) Group and Outline (all)
h) DataPilot

As an example: the currently disabled menu items in the Edit menu

As an example: the currently disabled menu items in the Insert menu
Use Case 6: Develop change tracking specification for the ODF standard

LibreOffice and Apache OpenOffice offer already today change tracking in ODF files. However, Microsoft Office has not implemented change tracking for ODF documents since Microsoft says the current change tracking specification with the ODF standard is defined insufficiently. Therefore this Use Case seeks the exact specification of change tracking within the next ODF OASIS and ISO standard in order to enable Microsoft to implement it in future versions of Microsoft Office.

The feature set is determined by the OASIS OpenDocument Sub Committee for Advanced Document Collaboration (CT SC). Currently the feature set defined by the deprecated Extended Change-Tracking (ECT) proposal of Microsoft has to be specified. It embraces the following items:

1. Development of conceptual methodology for text operations: meta components, meta operations, positioning components etc.
2. Definitions of operations for paragraphs: add, delete, move, format, merge etc.
3. Definitions of operations for text and headings: add, delete, move, format etc.
4. Definitions of operations for styles: insert and delete style, format paragraph and text, merge paragraphs etc.
5. Definitions of operations for lists: add and delete list items, merge paragraphs with list items, list-item addition followed by a merge with a succeeding paragraph etc.
6. Definitions of operations for tables: add and delete rows, columns, cells, cell content etc.
7. Definitions of operations for images, shapes, and charts: edit elements, change image references, change referenced images, edit text in shapes etc.
8. etc.

For a detailed feature set see the documents of the ECT proposal and the mailing-list of the CT SC.

Aside of to the specification work the following work items need to be executed:

1. Program an open source reference implementation of the new designed operations to provide a proof of concept for others to compare and guide their implementations and testing.
2. Participate in conference calls of OASIS (OpenDocument Technical Committee, OpenDocument Sub Committee for Advanced Document Collaboration) as well as ISO related organizations (e.g. DIN Gremium NIA-01-34, ISO SC 34 WG 6 and WG 5) and solve related problems on their mailing-lists.

The task is fulfilled as soon the change-tracking specification is part of the next ODF ISO standard. There are potential milestones where subpayments are negotiable:

1. Deliverance of definition of full feature set change-tracking to ODF TC
2. Deliverance of full feature set implementation of change-tracking as Open Source
3. Deliverance of ODF 1.3 OASIS Standard (including the change-tracking)
4. Deliverance of ODF 1.3 ISO Standard (including the change-tracking)

---

Tendering Procedure

Provider Details
The tender must contain a detailed introduction to the provider, its history, senior management, staff, their abilities and also reference projects and customers. Details on the number of staff, their background and experience (CVs) as well as previous reference projects (customers, scope of project, activities, etc.) are also expected. In addition, the tender must outline the skills and track record concerning LibreOffice and/or Apache OpenOffice software development, the OOXML and ODF specification, OOXML filter, etc. required for this project.

Method
The tender must describe the approach that will be adopted to implement the technical requirements. A solution concept setting out the activities, a schedule and development methodology is expected. Aspects such as testing, documentation, security and stability must be accounted for. Tenders must also specify how the resulting software source code at the end of the development stage will be incorporated into the main tree of the LibreOffice and the Apache OpenOffice open source project.

Language
Tenders and related correspondence must be written in German or English.

Deadline
Tenders must be submitted by to the Working Group spokesperson by e-mail and in PDF format by Wednesday, 30 April 2014. For any questions or comments the spokesperson of the Working Group Office Interoperability may be contacted:

Dr. Matthias Stürmer, stuerner@osb-alliance.com, +41 76 368 81 65 (Swiss mobile)
University of Bern, Institute of Information Systems, Engehalderstrasse 8, 2012 Bern, Switzerland