How Collabora Online development improves LibreOffice

By Jan Holešovský
Collabora Productivity

@Jholesovsky, kendy@collabora.com
Collabora Online

Rich, interoperable editing, everywhere
Collabora Online as a project

- https://collaboraonline.github.io/
- Development on GitHub: https://github.com/CollaboraOnline/online
- Translations at Weblate: https://hosted.weblate.org/projects/collabora-online/
- Forum for the users: https://forum.collaboraonline.com/
- And of course – LibreOffice as the core: https://www.libreoffice.org/
  - Improving LibreOffice improves Collabora Online
  - And improving Collabora Online improves LibreOffice
    - Not talking here about generic fixes
    - And covering only the last year (January 2020 – now)
Collabora Online uses tiled rendering

- The document is not painted as a huge area, but only as a set of 256x256 pixel “tiles”
- Handled by LibreOfficeKit
  - That is – LibreOffice core re-purposed to serve the tiles, events, and react on user’s input
  - Thin C/C++ API to access internals of the LibreOffice core
  - Focused on the tiled rendering
  - Callbacks for the events that come from the core – triggered by timers or users actions
The JavaScript part

- Initially was only a viewer
  - Extended to editing app (one editor)
  - Then shared editing – one editor who could edit, the others viewers
    - Was possible to switch who was editing, but still only one person
  - After several years – collaborative editing
    - Reusing the “multiple views” feature in LibreOffice (that I think nobody have been using until then ;-) - and suddenly it is the core of everything)
- Server part in C++, client part re-using LeafletJS
Collabora Online also reuses the dialogs

- Originally, to add functionality, various dialogs had to be re-written in JavaScript
  - Turned out to be unbearable long-term
  - Solution: Route the existing dialogs as pixels
- Dialogs not using the tiled rendering, invalidating only the relevant parts etc.
- Many corner case fixed
  - Like different users can have dialogs in different languages, or both open the same dialog type
  - Dialogs had to be made asynchronous
Rendering the core dialogs via JS

- Turned out that the sending the dialogs as pixels is bearable short-term, but has huge problems
  - Not theme-able much (ie. not beautiful)
  - Not fit for mobile
- Szymon Klos started implementing dialogs in JavaScript
  - That use the LibreOffice core dialogs definitions
- First for the mobile apps and online
  - To re-use the sidebar for navigation using one finger
  - Actions called via .uno commands

This is the sidebar!
The JSdialogs next development

- JS dialogs on the next level – now the Notebookbar is using the same thing
  - .uno: commands were not enough
  - The first implementation used the uitest API
    - Great – because improvements in the handling of JS dialogs improve the testing possibilities in LibreOffice and vice versa
- But even better now – using welding
  - Sharing the Gtk+ goodness
Improvements Since January 2020
General UI improvements

Collabora Online tries to reuse as much UI as possible from LibreOffice

- Various fixes in the .ui files
  - Eg. step increment, macros dialog, etc.
- Theming of the USE_VCL_PLUGIN=none case
- Conversion of many dialogs to asynchronous
- Rework the fontwork dialog to use IconView
- Various formula bar related improvements

Thanks mostly Henry Castro and Szymon Klos
Eclectic Notebook bar UX choices: to taste

Can collapse to just a menu ...
Notebookbar improvements

- Welded the Notebookbar
- Styles previews widget
- Crash fixes
- Various alignment fixes

Most of this thanks to Szymon Klos
Many improvements to PDF handling

Collabora Online uses the PDFium based importer in LibreOffice (where it is used for PDF previews)

- PDF search
- Annotations
- Handling of huge files
- Various PDFium fixes
- PDF export

(Tomaž Vajngerl, Miklos Vajna)

collaboraonline.com
Sidebar improvements

Used as-is for the desktop Collabora Online, and for the menu on mobile phones

- Improved synchronicity, lifetime, invalidation, ... (Michael Meeks)
- Instant update of various states (Szymon Klos)
- Default context improvements (Henry Castro)
- Various fixes in property panels (Henry Castro)
- Chart title & subtitle (Muhammet Kara, Kendy)
- FontColor command related fixes (Mert Tumer)
- Clear Direct Formatting (Mert Tumer, Pranam Lashkari)
- To Foreground / Background (Pranam Lashkari)
New .uno: commands added

Normally used primarily for scripting, and menus & toolbars definitions

- Added new setter parameters for many – like color etc.
  - And some commands were just a toggle, added an explicit parameter (Mert Tumer, Pranam Lashkari)
- Implemented several new ones around chart editing from sidebar
- Added several to context menus and other places
  - Where useful in the UI
- Various notifications improved (Ashod Nakashian, Pranam Lashkari)
- State changes notifications for various commands (Mert Tumer, Dennis Francis)
- Comment threading related .uno: commands (Pranam Lashkari)
Various corner cases & performance fixes

In many cases, various things are lurking unfixed in LibreOffice, but then triggered in Collabora Online much more easily

- Better handling of missing dictionaries (Szymon Klos)
- Performance improvements (Michael Meeks)
- Password protected files handling fixes (Miklos Vajna)
- SVG export improvements (Marco Cecchetti)
- Various DocumentModified state fixes (Kendy, Dennis Francis)
- Improved LanguageTag locale fallback mechanism (Kendy)
- Uittests framework improvements (Szymon, Kendy)
Collabora Online development improves LibreOffice

- Particularly the UI – Notebookbar, Sidebar
- But other areas too, eg. PDF support and scriptability
- And indirectly – interoperability etc.

By Jan Holešovský
@JHolesovsky, kendy@collabora.com