LibreOfficeKit – bridge between your application and LibreOffice

Szymon Kłos
Software engineer
szymon.klos@collabora.com
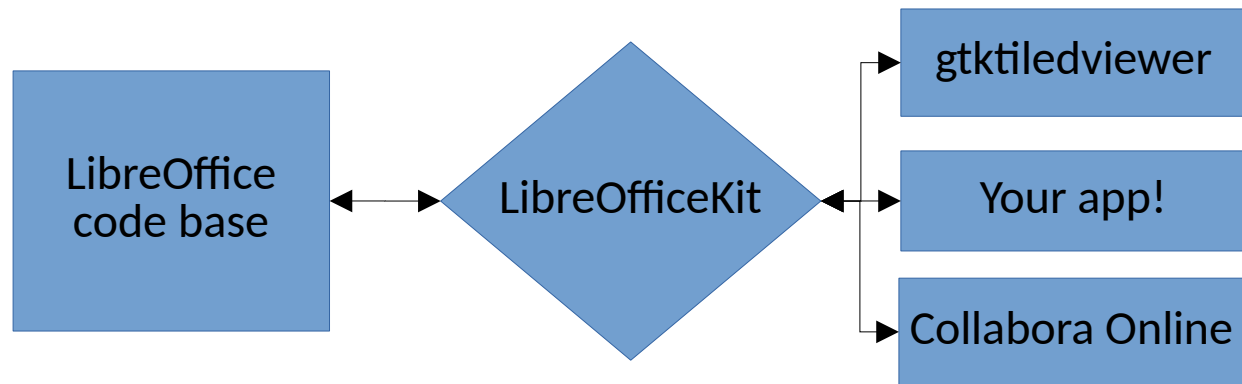
Collabora
Online

FOSDEM

LibreOfficeKit

Agenda

- What is LibreOfficeKit?

- Recent improvements

What is LibreOfficeKit?

- Bridge between LibreOffice goodnes and external world

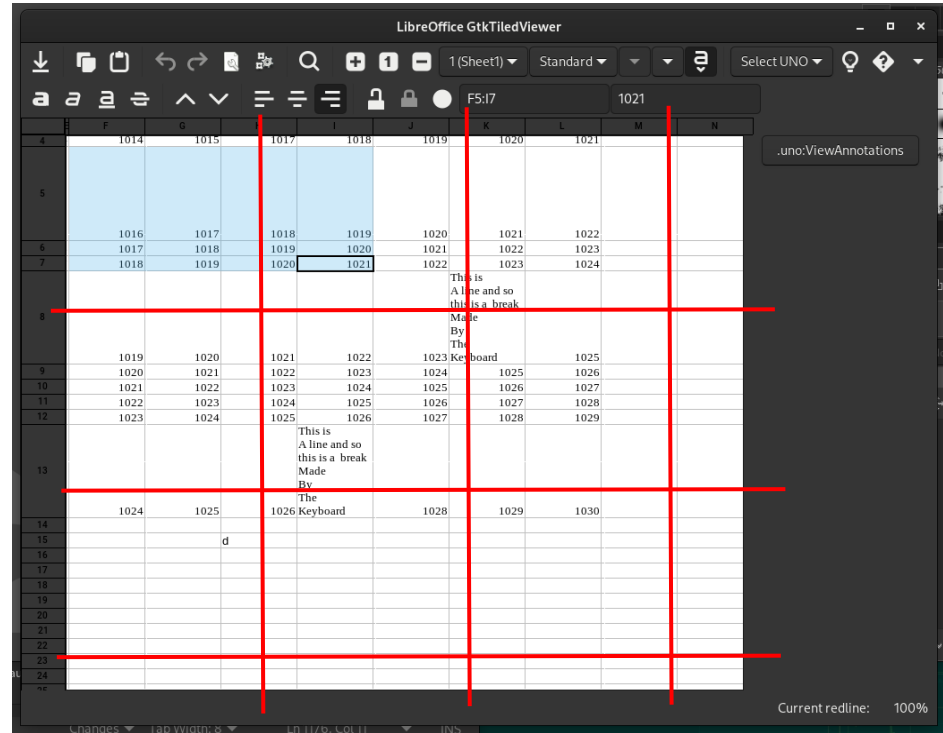- API providing tiled rendering, document manipulation, access to the UI components

```
                                              ┌──────────────────┐
                                         ┌───▶│  gtktiledviewer  │
                                         │    └──────────────────┘
┌───────────────┐      ◇             │    ┌──────────────────┐
│  LibreOffice  │◀───▶ LibreOfficeKit ◀───▶│    Your app!     │
│   code base   │      ◇             │    └──────────────────┘
└───────────────┘                         │    ┌──────────────────┐
                                         └───▶│ Collabora Online │
                                              └──────────────────┘
```

GTK tiled viewer

Example app

- **Renders tiled preview**

  of the document

- **Allows basic manipulation**

- **Calls UNO commands**


```
bin/run gtktiledviewer ---path=.../instdir/program .../spreadsheet.ods
```

LibreOfficeKit APIs

- Interfaces defined in **include/LibreOfficeKit/**

- Implementation in **desktop/source/lib/init.cxx**

- README in **libreofficekit/README.md**

- External code can invoke actions, render tiles using **LibreOfficeKitDocument** object.

- Some actions may trigger callbacks (tiles invalidation, selection changed), see possible callbacks in **include/LibreOfficeKit/LibreOfficeKitEnums.h**

- Callbacks usually return simple strings or JSON

Collabora Online

LibreOfficeKit APIs

- We try to keep LibreOfficeCode the same for tiled rendering mode, but sometimes it's impossible

- In such cases code is conditional, guarded by **comphelper::LibreOfficeKit::isActive()**

- Example: tiled rendering only draws the document's content – without selections. Selections are sent via LOK_CALLBACK_TEXT_SELECTION

Recent improvements

Master Page mode - rendering

- Problem: tile with the same number, with the same coordinates, but in different "editing mode"

- Two users can see the same slide in different mode and edit them as it would be independent piece of presentation
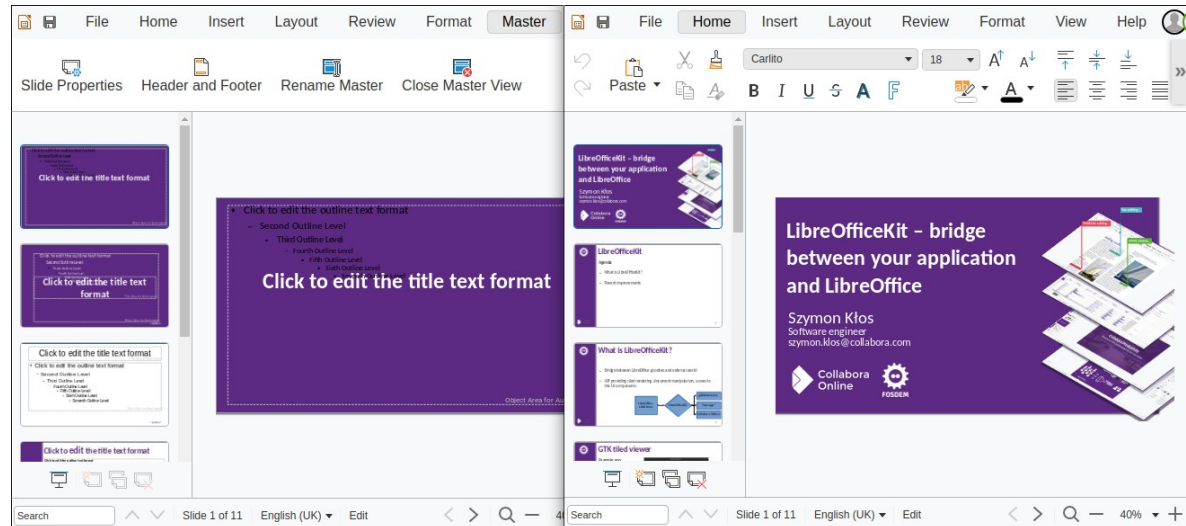
Master Page mode - rendering

- API for rendreing a tile was changed (added mode parameter)

- Algorithm which selectes view used for rendering selected tile was adjusted

- If there is available view in selected mode – use it!

- This allows to avoid unwanted side effects when switching edit mode

```
lok: masterpage: Introduce mode property to tile rendering
commit    931e3d204e59ccb2e9c479477e6e66b2c098924f

@@ -3770,6 +3790,7 @@ static void doc_paintTileToCGContext(LibreOfficeKitDocument* pThis,
 static void doc_paintPartTile(LibreOfficeKitDocument* pThis,
                               unsigned char* pBuffer,
                               const int nPart,
+                              const int nMode,
                               const int nCanvasWidth, const int nCanvasHeight,
```

Editing in-place cells with overflow

- Problem: when editing cell with overflow it was visible only in the editor's view. Any interaction from other users could render alternative version without edited field, what caused flickering.

Editing in-place cells with overflow

- We have to render in-place editing for all the views so tiles are always identical

- Edits are renderd always in the same order

- We also send cursor positions in the extended area what was missing before


```
lok: sc: render expanded EditEngine when editing in-place
commit      61d716901def607f01ab272996bf40b55642d650
```

# **Improved rendering of slide previews**

- When we render slide previews – we need to select some view

- Old implementation was using always first one

- Problem was that when first user was editing any text field

  - it was rendered for all users in the slide preview

- Algorithm for view selection was improved:

  we use any view which is non-editor with prefered view which
  already is watching the requested slide

```
lok: don't render active textbox on all slides
commit    81df11b1e5af610133874d9258999995a07364bc
```

Optimized iamge previews rendering

- Changed **doc_renderShapeSelection** implementation

- Previews are now rendered with maximum resolution: 1280x720

- This increased performance and UX when requesting the preview of large image embodied in the document

- Can be used while rotating

  an image
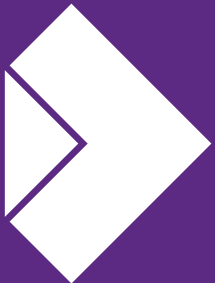

Elephant
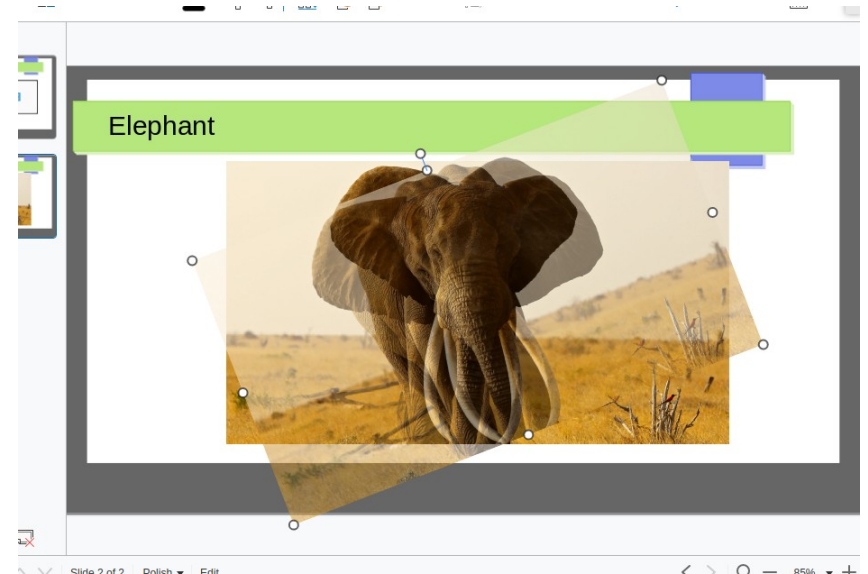
Exposed formulabar's content

- In the past formulabar was send as bitmap

- Now uses JSDialog – can be presented as native text field

- Events are sent as JSON with text content and selections

Thank you!

By Szymon Kłos

@CollaboraOffice
hello@collaboraoffice.com
www.collaboraoffice.com

Collabora Online