

LIBOCON19

Reusing the Online as an Android app

By Jan Holešovský

Collabora Productivity

kendy@collabora.com @JHolesovsky +holesovsky



History of LibreOffice on Android



2011-2012: “Get something on the screen”

Pioneered by Tor Lillqvist and Michael Meeks, LibreOffice was cross-compiled to Android

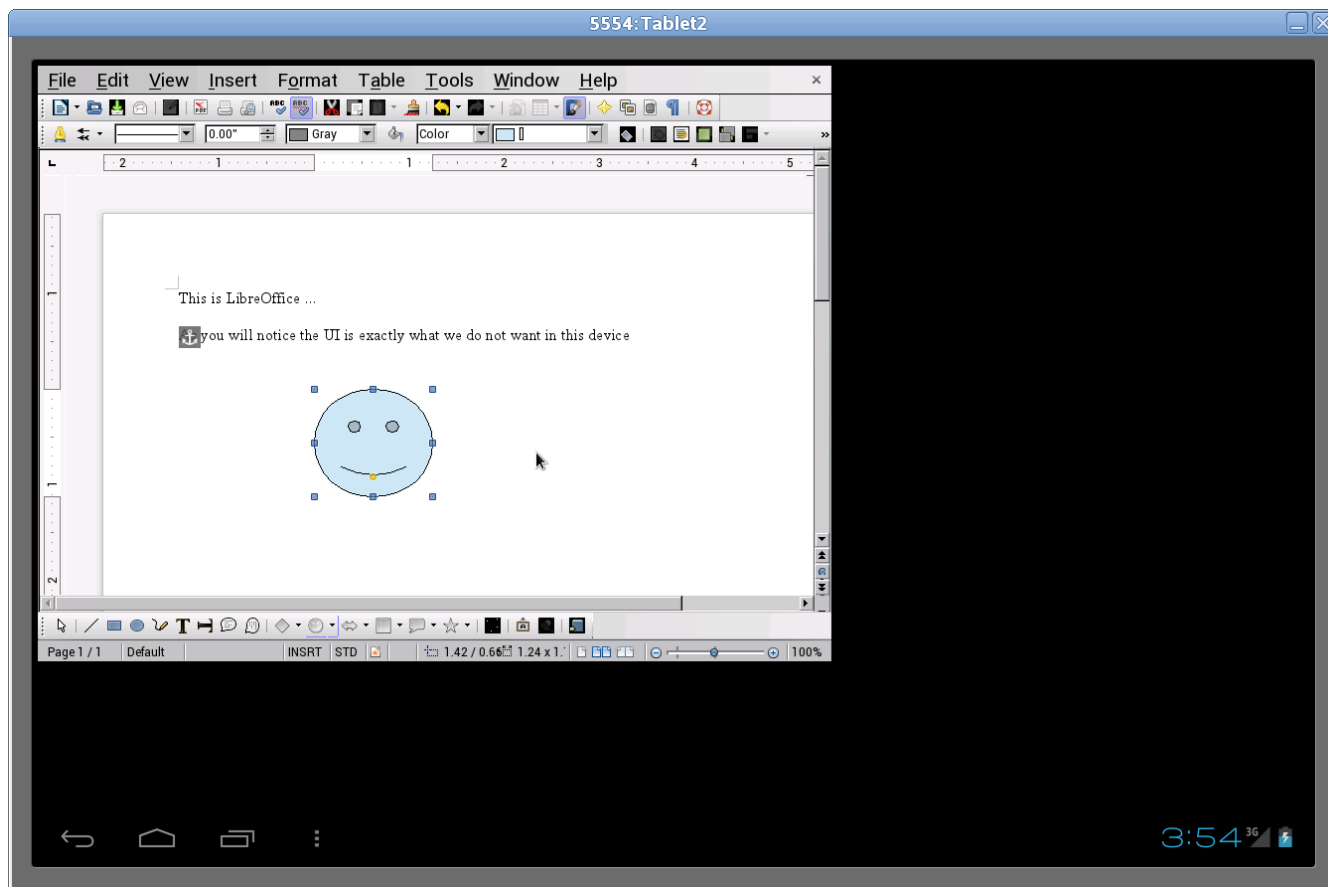
- Sheer amount of limitations on the way
 - Linker – a limited amount of dynamic libraries in one app
 - Fontconfig, using the assets, ...
 - Mergelib, components constructors – Matúš Kukan
 - First loader – Iain Billett
- Debugging a nightmare!
- But resulted in the first version showing LibreOffice on the screen



The 1st iteration just showed the entire LibreOffice UI

Not usable for real work,
but showed it was
possible!

- See the menus and toolbars

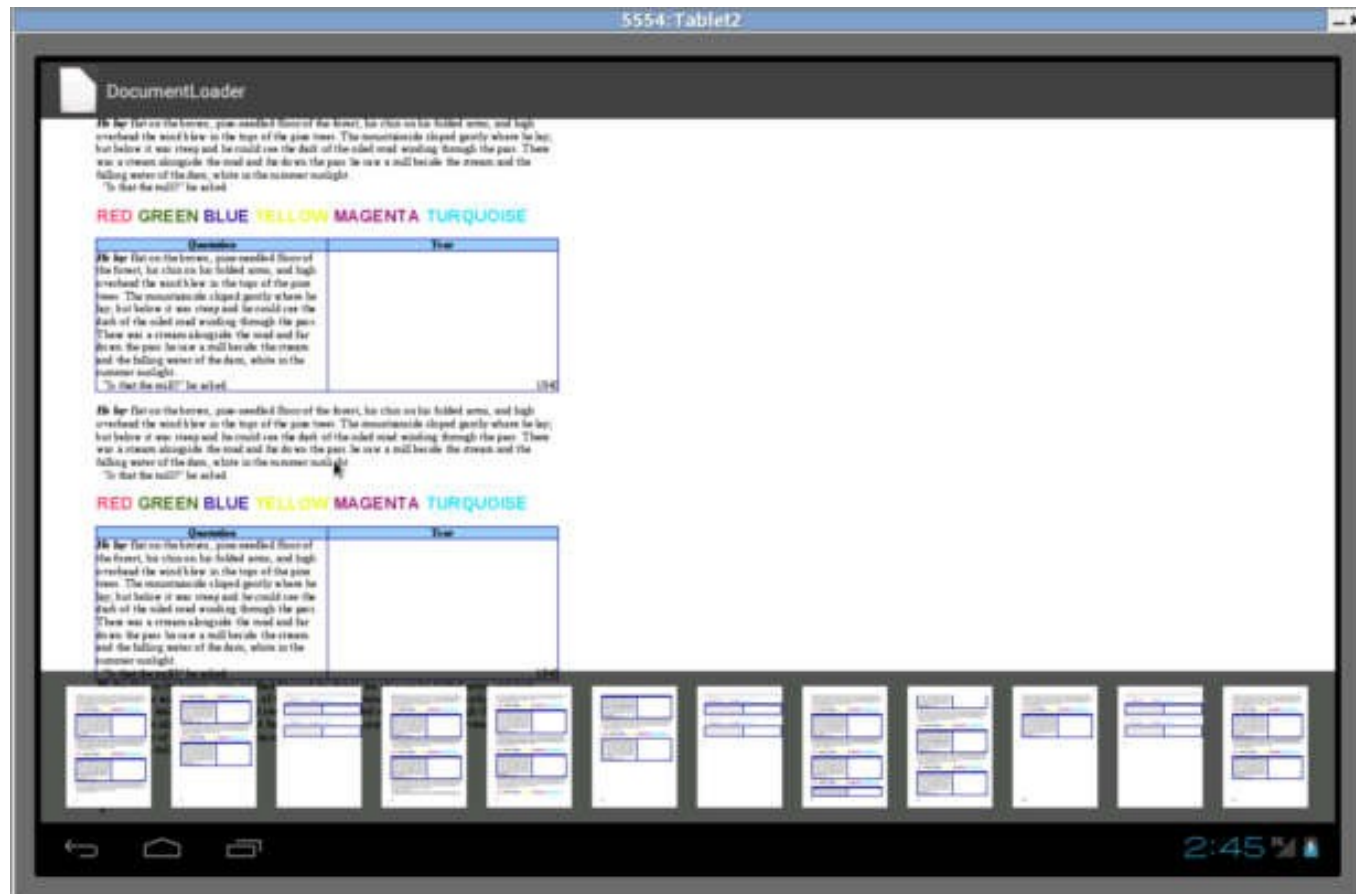




2012-2013: Rendering whole pages

Just a viewer

- But closer to what the mobile UI is supposed to look like
- Scrolling the entire document: problem

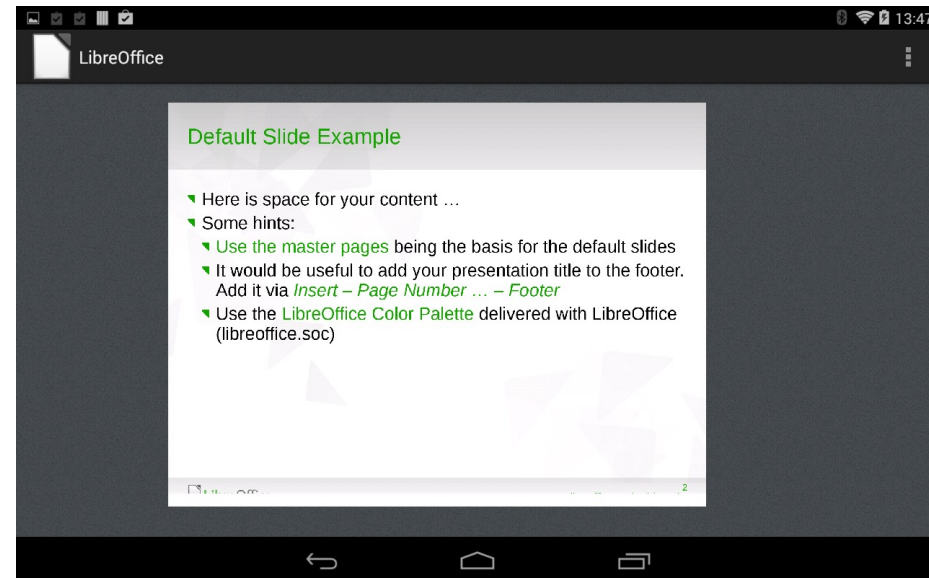




2014-2015: LibreOfficeKit used for rendering

Tomaž Vajngerl introduced tiled rendering into the Android app

- Using the compositing code from Mozilla
- The document consists of “tiles”, each can be rendered separately
- Editing made possible thanks to Miklos Vajna
 - It's easy to re-draw a tile after invalidation!





2016-2018: More features to the toolbar & elsewhere

Christian Lohmaier, Ximeng Zu, Mert Tumer and many more are adding various features

- But the Android app is not catching up
- Lots of LibreOfficeKit improvements triggered by the Online development





2019: Use the Online as the base for the Android app

Idea the same as what Tor Lillqvist has done for the iOS

- Make the app UI just a WebView
- Reuse loolwsd for the document rendering
- Use the existing JavaScript to compose the tiles, handle the UI, everything...
- Tor has prototyped that for iOS, and it worked, so why not on Android
 - Share code!

Reusing the Online code on Android



Loolwsd ported to Android

Based on the work that has been already done for iOS

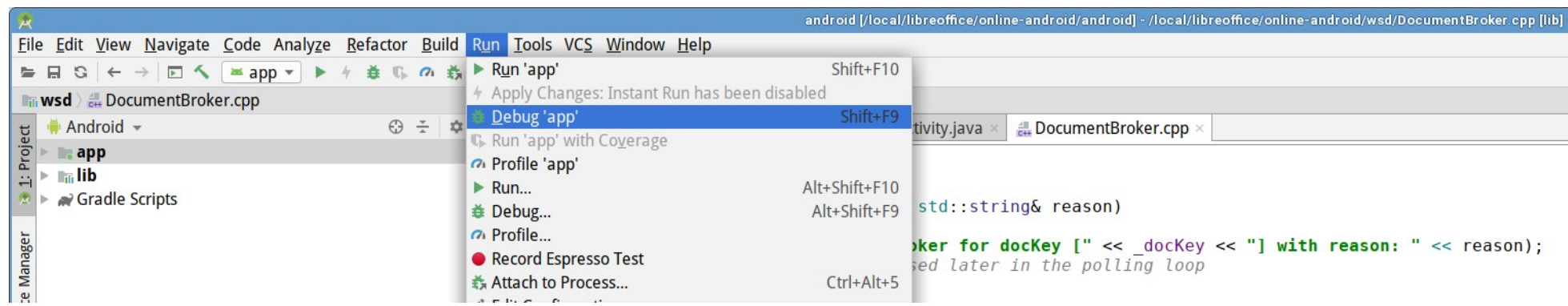
- Liblo-native-code.so had to be adapted so that it can be linked against
- Wanted the building the native code in online.git to be easy
 - Created an 'android/' subdir + the appropriate ./configure option
 - Then it should be entirely possible to build in Android Studio (modulo the liblo-native-code.so from the core.git)
 - Made the online.git part building via cmake Android Studio



Side-effect: Debugging the native part is easy

Works directly from the Android Studio

- You have to build the core.git with symbols
- But you want to limit their amount – otherwise linking of liblo-native-code.so takes ages
- See the details in online.git's android/README





Minimal app created

WebView over the entire screen

- Had to make the communication between the JS in the WebView and loolwsd working
- Using the FakeWebSockets the same way as the iOS app
 - JS → native: **`mWebView.addJavascriptInterface(this, "LOOLMessageHandler");`**
 - Methods on 'this' annotated with `@JavascriptInterface` can be called from the JS as `LOOLMessageHandler.<name>`, eg. `LOOLMessageHandler.postMessage(...);`
 - Native → JS:
`mWebView.loadUrl("javascript:window.TheFakeWebSocket.onmessage({'data': " + message + "});");`



Lots of functionality ported from the 'old' app

- The entire “LibreOfficeUIActivity” - that takes care of the initial document selection
- Associate the files with the editing activity
- Unpack assets that need to be read by the native part
- Displaying the License and the Notice
- Settings



New features via GSoC – thanks Kaishu Sahu

- **Print support:** One can print the document with the printer connected to the same network.
- **SlideShow support:** The user can view the document in the full-screen mode.
- **Insert Image in the document:** This allows to insert the image in the document from the phone's gallery.
- **Share document:** The user can use the existing messaging applications to share the document from the LibreOffice.
- **Save_As the document:** By this, one can make the fresh copy of the document anywhere in the storage media.
- **Rationale dialog for permissions:** To give the better understanding of a particular permission.
- **Launcher shortcuts support for api > 24:** The user can open the recent documents from the launcher (only if the launcher supports the shortcut feature).
- **Support for other document formats in save_as and share feature:** The user can choose the format of the document in which they want to share/save it.
- **Dimming the document when inactive.**



Bugfixing

Huge challenge: Fix the lifecycle

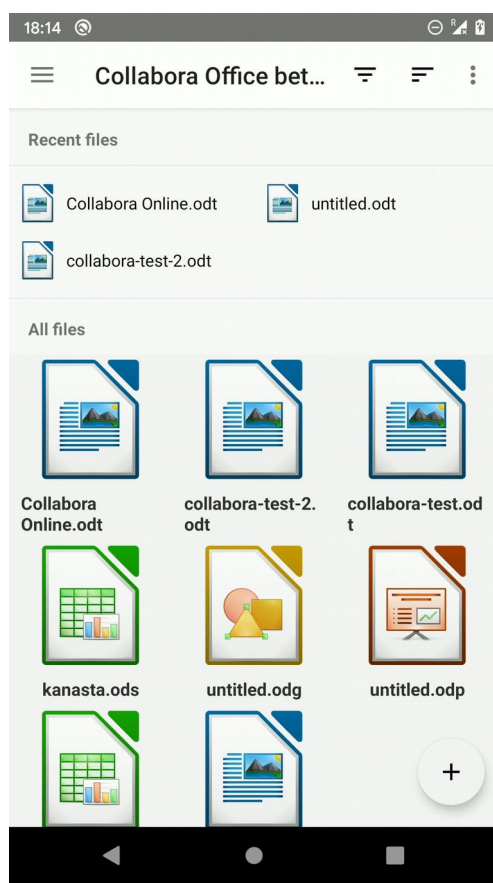
- So that switching to other apps work, without too much reloading / re-starting
- And that 2nd opening of the document work
- Ordering issue at the start; and more...

Various other fixes

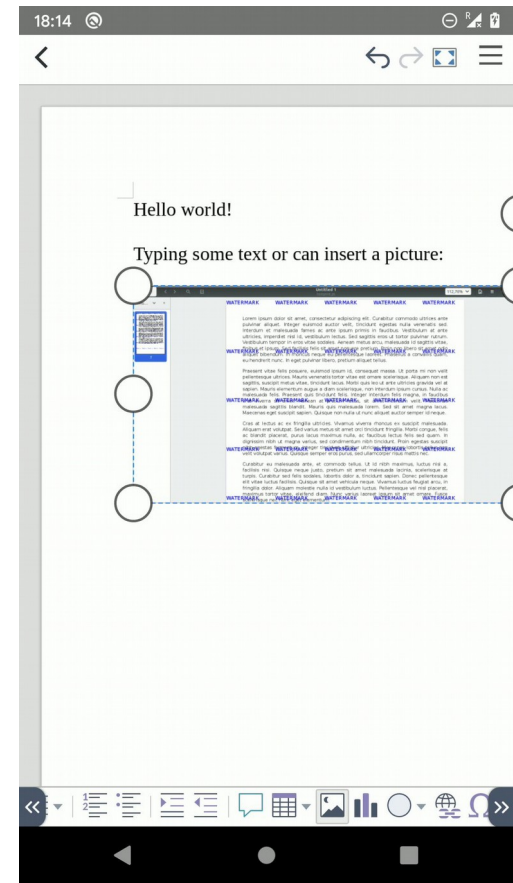
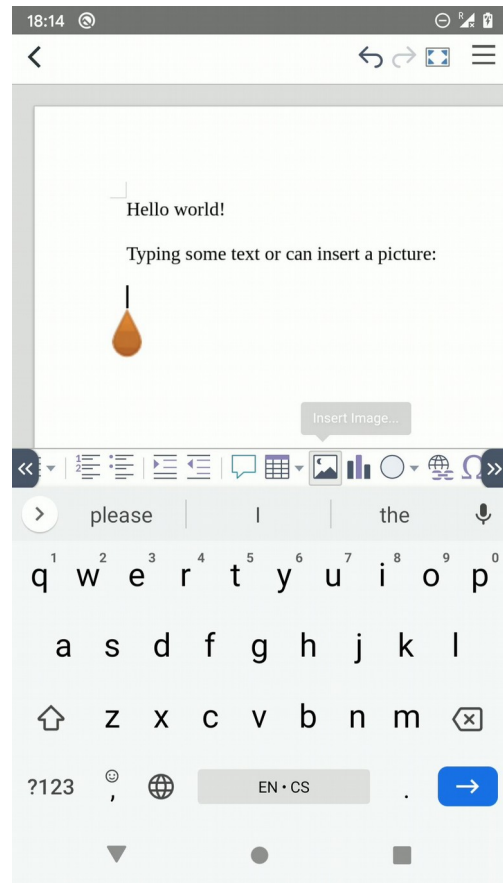
- Startup time – was huge thanks to the fontconfig / built-in fonts
- Some crashes, etc.



Demo (video)



collaboraonline.com



Reusing the Online as an Android app .. 16

Future steps



More to do

- Profile the text input
 - Suspect that the **WebView.loadUrl("javascript:...** takes a lot of time
 - If it's the source of slowness, should be possible to use a websocket to localhost
- Still sometimes the document does not load
 - And presents a “File not found” dialog box
- Update the document creation code
 - At the moment, we just copy a file to the new location; instead the TemplateSource property should be used
- Maybe few more fixes – but these are the most pressing



Collabora Productivity

Thank you – and get involved!

- Familiar development in Android Studio
- Steps how to build:
 - <https://gerrit.libreoffice.org/plugins/gitiles/online/+/master/android/README>
- Many thanks to: **Florin Ciornei, Gülşah Köse, Kaishu Sahu** – who committed to android/ in online.git

By Jan Holešovský

kendy@collabora.com @JHolesovsky +holesovsky