

# Work Report: TDF0002 – Milestone 1

## Development of Basic Editing Capabilities for LibreOffice for Android – Milestone 1

This project targets implementing basic editing in the LibreOffice for Android code. This first milestone demonstrates that the basic functionality required for the project is making good progress.

### Table of Contents

Development of Basic Editing Capabilities for LibreOffice for Android – Milestone 1.....	1
Overview of the task.....	2
2.1.1 (10.1.1) Integration of on-screen keyboard that is only visible in editing mode.....	2
2.1.3 (10.1.3) Triggering updates of views, Rendering and invalidation of tiles.....	3
2.1.2 (10.1.2) Routing Android events to LibreOffice app.....	4
Keyboard events.....	4
Touch Events.....	4
Cursor overlay.....	5
2.1.4 (10.1.4) Insertion and deletion of content.....	5
2.1.5 (10.1.5) Threading.....	7
In progress features.....	8
2.1.6 (10.1.6) Performance considerations, Support for OpenGL.....	8
2.1.7 (10.1.7) Rendering of selections on overlays.....	8
2.1.8 (10.1.8) Routing of application information from the program module’s core.....	9
2.1.9 (10.1.9) Texts and lists, Tables for Writer and Impress, Tables for Calc, Pictures and shapes .....	9
Development and Documentation.....	9
Project management.....	10
GtkTiledViewer - a LibreOfficeKit test-bed.....	10
Risks & Issues.....	11
Community engagement / public promotion.....	11
Conclusion.....	11
Appendix A: Commits Related to Deliverables.....	12
Appendix B: Commits Related to Android work.....	17
General Android bug-fixing.....	17
General Interoperability component inclusion.....	17
GtkTiledRenderer development test-bed.....	18

# Work Report: TDF0002 – Milestone 1

## Overview of the task

This Milestone was to deliver five of the relevant 2.1 items from the contract – including relevant items from 2.2 and 2.3. An analysis of five completed items, and status on all remaining items may be found below. Unfortunately due to a cut/paste problem the final consolidated contract re-numbered the SOW headings. Thus all headers marked 10.1.x are referred to as 2.1.x in the contract text. We provide both numberings in the heading hoping to clarify things in this regard.

The five items completed in their entirety for Milestone 1 are: 2.1.1 (10.1.1), 2.1.2 (10.1.2), 2.1.3 (10.1.3), 2.1.4 (10.1.4) and 2.1.5 (10.1.5).

## 2.1.1 (10.1.1) Integration of on-screen keyboard that is only visible in editing mode

We provided a way to trigger the visibility of the on-screen keyboard. By default, it is not visible, so the editor acts as a viewer. However, as soon as the user taps inside a paragraph, the on-screen keyboard appears. When the “down” button below the on-screen keyboard is pressed, it gets hidden. We added a LibreOfficeKit callback mechanism to LibreOffice core, so that when in the future we add support for e.g. selection of graphic objects where showing the on-screen keyboard does not make sense, we can easily notify Android when to show/hide the on-screen keyboard automatically.

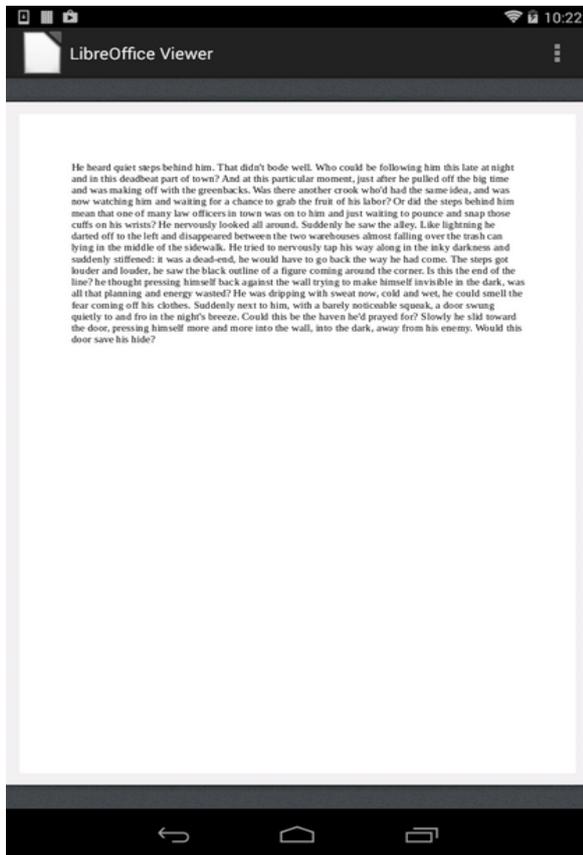


Figure 1: Default, viewer mode: no on-screen keyboard.

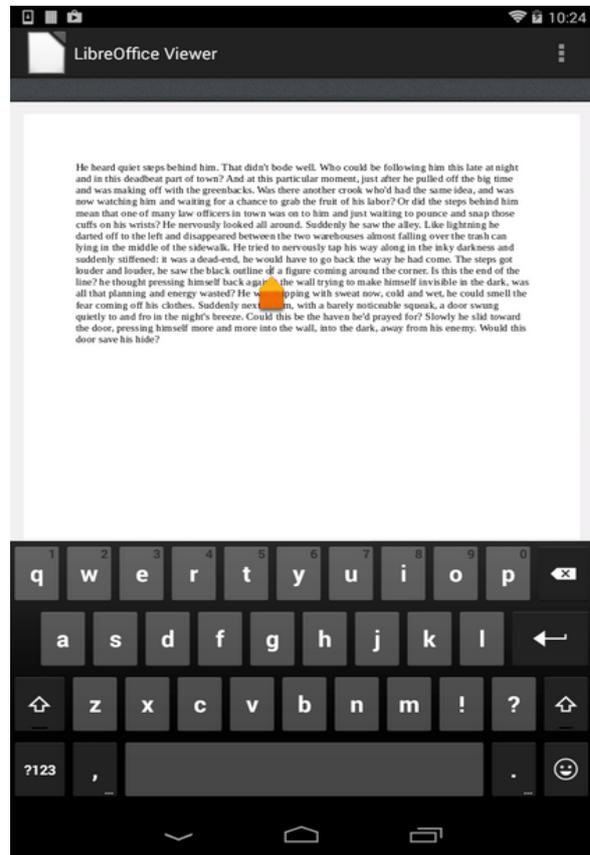


Figure 2: On-screen keyboard is shown after the user taps inside a paragraph.

# Work Report: TDF0002 – Milestone 1

## 2.1.3 (10.1.3) Triggering updates of views, Rendering and invalidation of tiles

Communication between Android and LibreOffice is a two-way channel. To avoid implementing both at the same time, first we went after the LibreOffice → Android direction. This means that while the previous assumption was that once a tile is rendered, it can be kept in a cache on the Android side forever – due to editing, we need some way to invalidate a subset of that cache as part of the document is changed.

The desktop rendering uses a mix of direct painting and “invalidation, then repaint using a timer”. The expected approach would have been that we always just invalidate an area of the window when we want to repaint, and then the main loop would decide when to repaint. Unfortunately at the moment our main loop has no priorities (this is being worked on in the feature/priorities branch by Munich students), so when we need low-latency repaints on the desktop, we must paint directly for now. This is a problem for Android, where we want to subscribe to invalidations and repaint the Android view accordingly.

We solved this problem by making the applications aware of being used in tiled editing mode, and if that's the case, then we always invalidate the to-be-repainted area, never paint directly. This required code changes in both Writer and EditEngine.

The other problem is how to notify Android about what tiles to throw away, as LibreOffice doesn't know what tiles Android has. Android just asks for tiles, LibreOffice gives them away, but their lifecycle is managed by Android. We solved this by sending a rectangle of the invalidated area to Android (in document model coordinates), and then Android evaluates what cached tiles intersect with the area, and discards those.

Using this technique, we extended the `vcl::Window` subclass of Writer, Impress, and Calc to emit such invalidation events to Android.

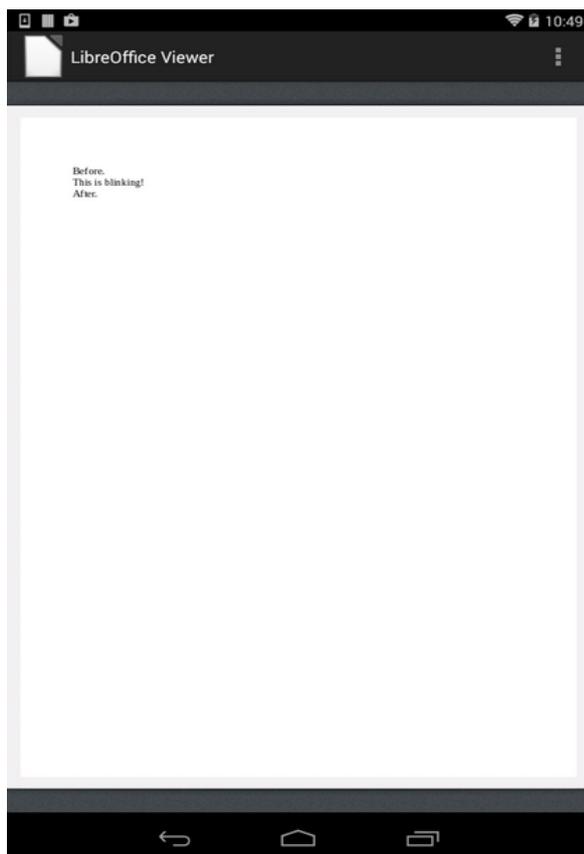


Figure 3: Blinking text -- second paragraph is visible

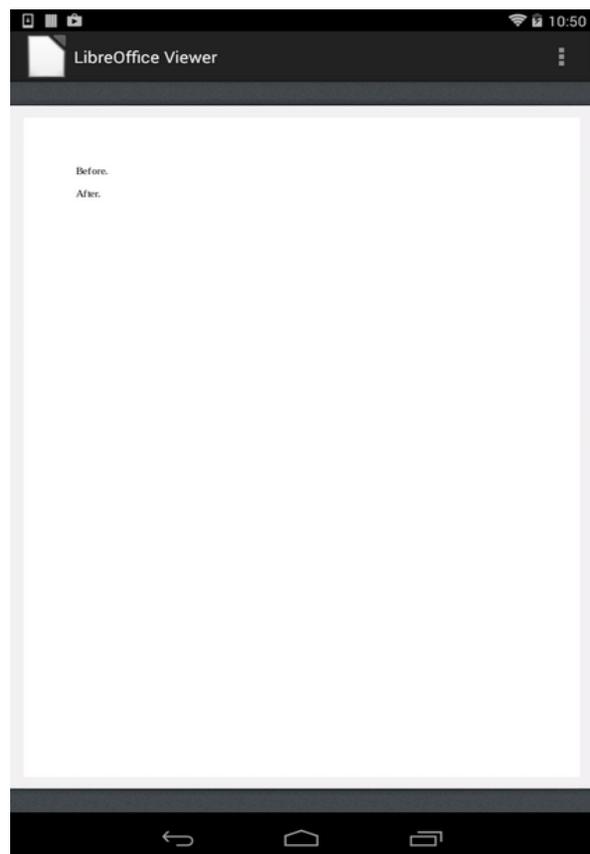


Figure 4: Blinking text -- second paragraph is hidden

You can see this in action at <http://youtu.be/EkGpCrG1h1Y>.

# Work Report: TDF0002 – Milestone 1

## 2.1.2 (10.1.2) Routing Android events to LibreOffice app

Sending events in the Android → LibreOffice direction is a bit more complex, as LibreOffice is a desktop application, so we had to find a way to map the events of the on-screen keyboard and the touchscreen to events expected by LibreOffice (keyboard, mouse), and extend LibreOffice when such mapping was not possible. There were 3 major sub-tasks here: routing keyboard events, routing touch events and cursor overlay.

### Keyboard events

Both Android and LibreOffice can handle real characters by their Unicode integer representation. However, for special keys, we need to map between them, as both Android and LibreOffice only provide constants for them. We added mapping for the most frequent special keys like backspace and enter. The mapping is done on the Android side, the constants of LibreOffice are available to Android using the Java UNO binding. Then the LibreOfficeKit implementation of the keypresses simply sends the event to the currently active frame – the benefit of this is that it works out of the box in all applications.

### Touch Events

We map touch events to LibreOffice's VCL events. LibreOffice sees a long tap as a doubleclick and so on. Where that was not possible, we extended core to recognize the touch gestures. For example the desktop application did not allow adjusting the *start* of a selection – the new `lok::Document::setTextSelection()` API allows to do that. The other problem is that Android sends the coordinates of the event in document model units (i.e. it's independent from the view state of the document), but LibreOffice expects to get them in screen pixels.

Mapping from document model coordinates to screen ones would not be clean, the pixel position depends on the size of the menus, toolbars and other controls in the desktop application. Instead, we extended core to allow sending the coordinates in document models when doing tiled editing. This is currently implemented in Writer and Impress, but we plan to do the same for Calc.



Hello world! :-)  
Multiple lines.



First edit. Hello world! :-)  
Multiple lines. Second edit.

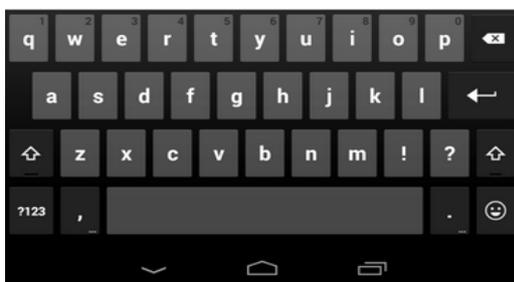


Figure 5: Multiple lines typed into an empty document.

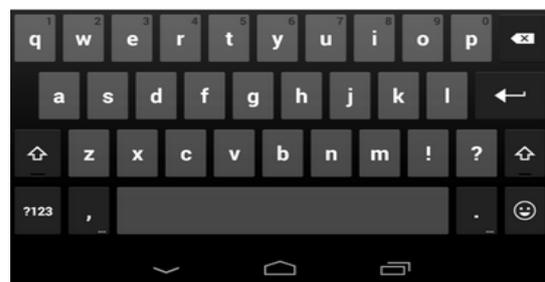


Figure 6: Touchscreen used to reposition the cursor: so not only appending content at the end of the document, but also insertion at other positions is possible.

You can see this in action at [http://youtu.be/4CJbu\\_QNUh0](http://youtu.be/4CJbu_QNUh0) and <http://youtu.be/frQWyjjsl3I>.

# Work Report: TDF0002 – Milestone 1

## Cursor overlay

As can be seen in the above screenshots, when the editing mode starts, we display a blinking cursor at the position where LibreOffice thinks the cursor is. It's a rectangle with a rather small width. The LibreOfficeKit callback introduced originally for invalidation of tiles is reused for this purpose, too: it tells Android where the cursor rectangle is. To avoid repainting the affected tiles all the time when the cursor blinks, we implemented a transparent overlay on the Android side that paints the cursor rectangle and its handle. The handle at the moment is not too useful, but we plan to allow dragging it, and use similar handles later for adjusting the selections.

## 2.1.4 (10.1.4) Insertion and deletion of content

Once both tile invalidation and keyboard events have been implemented, we expected that both insertion and deletion of content would work out of the box. This was true for insertion, but sadly not for deletion. The problem was that deletion (the backspace key) is mapped to an application-specific UNO command (e.g. `.uno:SwBackspace` for Writer) and we wanted to have it working in all applications, without hardcoding the key event integer code → UNO command name mapping. We solved the problem by enabling the sfx dispatch framework in the Android build as well, then once all the required components were available, deleting worked as well.



Figure 7: Insertion of content in Calc



Figure 8: Calc showing the calculated formula result

# Work Report: TDF0002 – Milestone 1

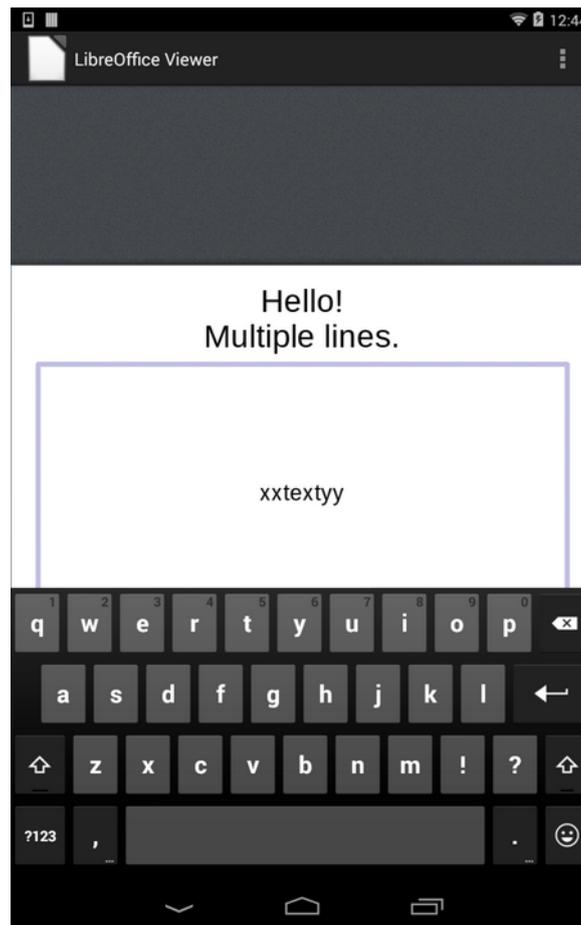


Figure 9: Text in multiple Impress shapes

## 2.1.5 (10.1.5) Threading

Threading had to be introduced as there are two areas where a single thread is not enough. The Android application is a normal Java Android application, except that it also ships the LibreOffice codebase as a single shared library: `liblo-native-code.so`. We use the LibreOfficeKit JNI bindings of its C API to interact with LibreOffice from the Java app. The app implements editing support using 4 threads:

1. The Android UI thread, we can't perform anything here that would take a considerable amount of time.
2. An OpenGL thread which contains the OpenGL context and is responsible for drawing all layers (including tiles) to the screen.
3. A thread (we call it `LOKitThread`), that performs LibreOfficeKit calls, which may take time. It also receives events from the `soffice` thread (see below) when the callback emits an event.
4. A native thread created by LibreOfficeKit (we call it the `soffice` thread), where LibreOffice itself runs. It receives calls from `LOKitThread`, and may emit callback events as necessary.

When we introduced tile re-rendering we noticed slower and sometimes unpredictable behavior at tile redrawing. There were three reasons for this:

First is the use of `PriorityBlockingQueue` for the queue implementation in `LOKitThread` for queuing of events. The problem with `PriorityBlockingQueue` is that it doesn't guarantee the order of elements with the same priority is preserved so the effect of this was that tiles were drawn in reverse order. We could solve this by adjusting the compare function, but instead we exchanged the implementation with simpler non-priority `LinkedBlockingQueue`.

# Work Report: TDF0002 – Milestone 1

Second reason was `CopyOnWriteArrayList` for storing tiles, which is a synchronized `List` implementation that doesn't need special locking. It turned out that the problem with `CopyOnWriteArrayList` is that it takes some time before other threads sees the change that was made in another thread. The effect of this was that when new tiles were added by `LOKitThread` and a redraw was requested to the OpenGL thread, the OpenGL thread did not always see the newly added tiles thus it did not paint the new tiles on the screen. This problem was solved when we exchanged the `CopyOnWriteArrayList` with simple `ArrayList` and introduced `ReadWriteLock` mechanism. With `ReadWriteLock` we can lock all access (read and write) to the `ArrayList` when we do a write operation and still allow threads to read from the list in parallel.

The third optimization done was to minimize the amount of events and do more in `LOKitThread` instead of UI Thread. Previously, when a change to the document viewport was done (for example the user was scrolling the document) the UI thread calculated if new tiles were needed and if yes, the event for every new tile was sent to `LOKitThread`. This calculation is not simple and can cause synchronization problems between threads (UI and `LOKitThread`). In addition to that, the amount of new tiles requested could fill up the queue (when the user is scrolling quickly through the document), and at the time the tile was processed it could already be out of document viewport. Instead of this approach we relieved the UI thread of the calculation and just send `LOKitThread` the signal that the viewport changed. The `LOKitThread` now calculates and adds new / removes unneeded tiles in one event. Additionally the tile invalidation is done similarly in one event in `LOKitThread` too.

Regarding locking the data used by multiple Java threads, we follow the Android guidelines. When `LOKitThread` calls into `LibreOfficeKit`, then we invoke native code that runs on that thread, so we again need locking to avoid race conditions with the `soffice` thread. We solved that problem the same way as UNO method implementations work: at the beginning of each such `LibreOfficeKit` method, we lock the solar mutex (the big global lock of LibreOffice).

## In progress features

These features are not part of the Milestone 1 Deliverables, but give an insight into the progress made towards the final milestone for the sake of interest.

### 2.1.6 (10.1.6) Performance considerations, Support for OpenGL

Thus far performance appears to be acceptable, and we have a partner engaged in doing some profiling to determine if there are low hanging fruits that can be improved. Unfortunately the OpenGL work while rendering nicely on PC has some significant and hard to debug lifecycle problems the fixing of which has tied up resource that could otherwise have been used on mobile.

### 2.1.7 (10.1.7) Rendering of selections on overlays

This feature is in progress, we have finished the core part of it, and the Android side just added rendering support for text. Here is how it looks like in `gtktildeviewer`:

# Work Report: TDF0002 – Milestone 1

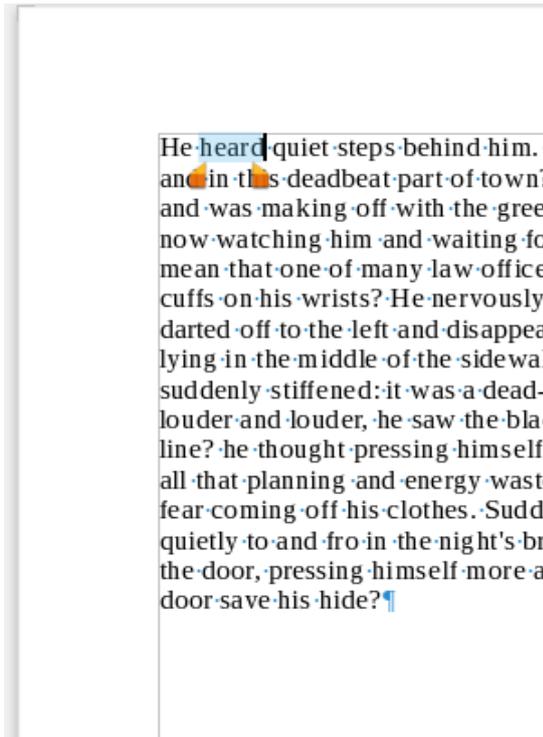


Figure 10: Semi-transparent selection overlay in gtktildeviewer

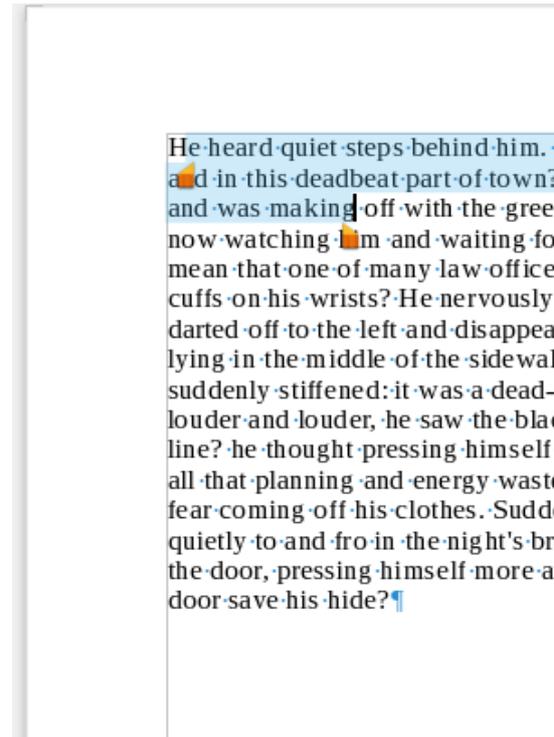


Figure 11: Both ends of the selection adjusted in gtktildeviewer

## 2.1.8 (10.1.8) Routing of application information from the program module's core

As explained in the tile invalidation section, we now allow the Android client to register a callback via LibreOfficeKit into LibreOffice. Using this, application modules can send information to the client. At the moment the following commands are sent this way:

1. Invalidate tiles
2. Invalidate position and size of the visible cursor
3. Invalidate text selection (position and/or size)
4. Invalidate start of the text selection (position and/or size of the selection start handle).
5. Invalidate end of the text selection (position and/or size of the selection end handle).

## 2.1.9 (10.1.9) Texts and lists, Tables for Writer and Impress, Tables for Calc, Pictures and shapes

This is in progress:

1. Text and lists are implemented.
2. Tables, pictures and shapes still needs work.

## Development and Documentation

Commits implementing the above features are available in the master branch of LibreOffice's core.git repository. For each new method we introduced, we added doxygen documentation. Higher level documentation is available at the following places:

1. Building and debugging information is available at the much updated README.Android file in the

# Work Report: TDF0002 – Milestone 1

core.git repository.

2. The architecture of the tiled editing approach is explained in the “LibreOffice on Android” FOSDEM talk:



Figure 12: Ctrl-Click on the picture to access the slides

3. Documentation about LibreOfficeKit is available in the libreofficekit/README file, that we update in parallel to our changes.

## Project management

We communicated our progress with the project in the form of blog posts. When a feature was finished in both core and on the Android side, we published a blog post. Here are the so far published entries:

1. <http://vmiklos.hu/blog/tiled-editing-part-1.html>, titled “Tiled editing: from viewing only to a living document”
2. <http://vmiklos.hu/blog/tiled-editing-part-2.html>, titled “Tiled editing: from a living document to input handling”
3. <http://vmiklos.hu/blog/tiled-editing-part-3.html>, titled “Tiled editing: from input handling to selections”

We plan to continue to provide project management & progress updates in the form of technical blog posts, linking the code to the relevant functionality, with videos etc.

## GtkTiledViewer - a LibreOfficeKit test-bed

Recognizing the difficulty of development and debugging under Android, as well as the long development iterations required, we have implemented (cf. Appendix B) a gtk+ based tiled rendering harness that allows us to rapidly prototype and test new LibreOfficeKit API under Linux. This is rather a successful move, as it allows a development iteration of a few seconds, while uploading a new .apk to the device usually takes at least one minute – you can see a screenshot here; it is now possible to do basic document editing using this.

# Work Report: TDF0002 – Milestone 1

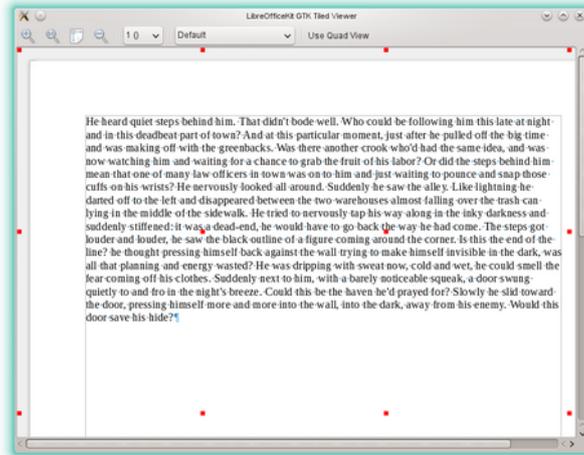


Figure 13: The gtk+ tiled viewer, running on Linux, showing the top left corner of each tile.

It is hoped that this will also provide a means to re-use the core of LibreOffice to build custom editors in future – something that has been of interest in the past.

## Risks & Issues

The editing work was commenced based on an understanding that Calc's tiled rendering was already working. Unfortunately, in part related to unexpected internal personnel movement, this still is only partially so. Calc is the weakest piece of our tiled rendering and hence editing story. While the basic tile vertical positioning still presents problems on interactive zoom on Android, we have implemented basic editing although this is still limited to the first cell only. We continue to work on fixing the underlying Calc tiled rendering issues.

## Community engagement / public promotion

We produced three detailed blog posts about the ongoing work – please see the above links from Miklos, however, unfortunately thus far no community members have got involved in the coding. There is interest from one developer in a Google Summer of Code role in turning gktiledviewer into something more powerful for use in GNOME Documents.

## Conclusion

We hope that our write-up highlights the current state of Android editing. Progress is rapid, with a large volume of not only editing commits, but other work related to improving the existing viewer from a performance, code-size, and stability perspective.

One remaining issues is around test-ability of the editing functionality, naturally while APKs are available from the tinderbox, there is no direct channel for users & testers to play with this on Android. Collabora explicitly disables editing inside the Android Viewer. It would be good to agree some means to enable users to test and get excited about the Android editing functionality, either TDF providing its own builds with that enabled, or some suitable dialog / crediting to enable it in Collabora's Beta builds. Feedback appreciated.

We really appreciate the opportunity to work with The Document Foundation, and to contribute significant value to the LibreOffice project together, and are looking forward to continuing the work, to complete it for Milestone 2 in around another month.

# Work Report: TDF0002 – Milestone 1

## Appendix A: Commits Related to Deliverables

Commits are listed with their short hash from master, in chronological order.

<i>Commit ID</i>	<i>Commit message</i>
6123d6a	android: Add initializeForRendering to LOK
83e84b9	android: take document borders into account for document size
2005e34	android: set single page mode in writer when init for tiled render
bea2ade	vcl::ITiledRenderable: add registerCallback()
0137974	vcl: add OutputDevice::LogicInvalidate()
aa827f5	lok::Document: add registerCallback()
3547424	sw: notify LOK clients about invalidations
a989d6f	LOK: add LibreOfficeKitCallbackType enumeration
85ce7c0	Factor out XFillBitmapItem::dumpAsXml() from sw
f4fadeb	scaddins: fix Android build
0585030	jvmfwk: fix Android build
5a94bc9	drawinglayer: fix Android build
db8ac4f	javaunohelper: missing include
27a644b	lok: Add comments to some methods, put the star where it belongs
0dbc6f5	android: No reason to return JNI version less than Java 6
3b77cd3	android: Add support for callbacks from LO to JNI and Java LOK
8de3943	Add lok::Office::postKeyEvent()
e905d3b	android: interface for processing of LOK messages
a04e325	android: Process messages in TileProvider + invalidation interface
616a0ac	android: add toString to TileIdentifier
156411e	android: Support force redraw in "tile request" event
57ffd85	android: react to tile invalidation request and invalidate tiles
b214b0b	android: Add inverseScale to RectUtils
b6a1207	android: Fix what invalidate rect numbers should mean
2755cc0	android: allow TileLayer to expose the image
2cd20fa	android: support to rerender tile buffer directly
8da6499	android: bypass Bitmap creation and copying when creating tiles
e9c6949	android: reuse rerenderTile in createTile
0ecf41b	android: remove checkerboard color property - fix set to white
9f77740	android: force screen to render - add access to LayerClient
723b6a0	jni: add postKeyEvent to Office and redirect the call to LOK
4f1eef2	android: show version / buildid in about dialog
47287df	libreofficekit: typo
2d92cfd	make OverlayTriangle public, allow to change each point
08d7111	android: comments for new interfaces and methods
fca89ae	LOK: add LibreOfficeKitKeyEvent enumeration
1b75735	android: remove logging null in DirectBufferAllocator
93af546	jni: if documentLoad returns NULL don't call NewDirectByteBuffer
6dabfea	android: remove unneeded comment & clean whitespace
5291c63	android: add method to force showing of soft keyboard
5c3c862	Missing guard in SwXTextDocument::initializeForTiledRendering()
56dd27d	sw: invalidate after keypresses when we have a callback
f1a2759	android: if we don't have a document, then that can't be text
9624077	android: avoid setMessageCallback() for now
f260218	android: Inflate viewport even more to render more tiles offscreen
99db6d8	SwRootFrm::CheckViewLayout: never center pages when doing tiled rendering
9069456	SwViewShell::PaintTile: paint a small red rectangle as well in debug mode
0f291b3	android: remove CheckerboardImage class as it is not needed
da39c5f	android: fennec import - double tap zoom in JavaPanZoomController
a3efafe	android: add support for touch event to LOEvent
ef73f53	android: show software keyboard on main view in main thread
cb85026	android: send touch events in JavaPanZoomController
1e4f6ee	android: show soft keyboard on single touch event
a5e2c36	android: guard buffer allocation and return null if alloc. fails
7cb282b	android: provide context to ComposedTileLayer

# Work Report: TDF0002 – Milestone 1

4712db6 android: add onTrimMemory to ComposedTileLayer  
24a92a4 android: simplify methods with currentZoom / currentViewport  
6b71410 android: create tile outside of beginTransaction  
548ed2c SwViewShell::InvalidateWindows: when tiled rendering, ignore visual area  
a81010b SwViewShell: make sure no callback is fired during PaintTile()  
d6a1db7 android: register invalidtion callback on first touch  
db21465 SwViewImp::AddPaintRect: when tiled rendering, ignore visual area  
56e7490 android: remove invalidation registration on touch for now  
b746908 android: add key\_press event and send them to LOKitThread  
3766397 android: add missing break  
389fae7 android: Add some comments.  
89d4175 android: Add needed services for editing.  
bd065f5 android: Add key events in TileProvider & pass key events to LOK  
3cb66f5 android: move getMemoryClass to LOKitShell  
29d65fb android: remove getDisplayMetrics as it exists in LOKitShell  
96fa96b android: cleanup GeckoLayerClient  
d1aad61 android: fix invalidation rect calculation in ComposedTileLayer  
e053672 libreofficekit: update README, deploy a couple of g\_info() calls  
462d48c android: make package name configurable  
d4ebde6 android: re-enable calc documents associations with LO viewer  
07997cb android: adapt doc browser to updated manifest that accepts Calc documents  
0869d8e Add initial lok::Office::postMouseEvent()  
166df60 android: Remove 200kb docu.png from resources as it is not used  
f34f51b android: eliminate hardcoded LibreOfficeMainActivity and package  
7aad617 android: don't need Bootstrap anymore  
b66690b android: release local ref to string after callback exec. in JNI  
459979b libreofficekit: cosmetics  
61c368d android: cleanup new tileid's if they gone out of viewport scope  
70766d8 android: construct a "css" rect of the tile in TileIdentifier  
f3fdc82 android: small cleanup of ComposedTileLayer  
07a736f android: add getCSSRect to TileIdentifier and use id in SubTile  
5ad4180 android: make editing switchable via LOKitShell  
3135fbb android: invalidate only new/changed tiles - big perf. improvement  
3596abe android: getRect renamed to getRectF - adapt the code to this  
784a176 android: guard rendering if buffer could not be allocated  
ef99aee android: eliminate hardcoded LibreOfficeMainActivity and package in LOAbout  
94ec99d android: fix crash in tile provider while closing the document  
167412c android: Add makefile rule to sign the resulting .apk.  
bb0b878 LOK: move postMouseEvent to Document  
752d151 Add SwXTextDocument::postMouseEvent()  
3d4c1b9 Initial SwEditWin::LogicMouseDown/Up  
52246e0 SwXTextDocument::postMouseEvent: missing guard  
79317f2 android: Fix loading files that have space in the filename  
4c9030b android: it doesn't hurt to log the document filename on load  
3928932 android: 'make release-apk' now also accepts the example document.  
dbde7f6 android: add postMouseEvent to LOKit JNI interface  
718e46c configure: fix --with-android-package-name wrt ndk-gdb  
0046957 android: disable Impress View -> Slide Pane  
a791560 android: make sure the soffice.cfg directory is always available  
c1d2c39 android: use getUnicodeChar() in LOKitTileProvider  
d0b5023 android: translate DEL/ENTER in LOKitTileProvider  
8fa9f14 android: add some comments to TileProvider interface  
41e5824 android: add mouseButtonDown(Up) to TileProvider + impl.  
525d73f android: extend touch event with transformed document coordinate  
06da738 android: view point to document point conversion updated  
f6e82e2 android: convert to document coor. and send it with touch event  
e57a649 android: touch pop-ups keyboard and sends mousebuttondown event  
eb2de5c LOK: don't ignore all callback events when we're in the callback already  
c99e634 tdf#80960: Attempt to fix the cursor placement after document load.  
ec170c9 Add LOK\_CALLBACK\_INVALIDATE\_VISIBLE\_CURSOR  
b83b96c SwVisCrsr::Show: ignore VisArea() in case of tiled rendering

# Work Report: TDF0002 – Milestone 1

43bbf53 android: remove unneeded code from GeckoLayerClient  
dd6e502 android: move TouchEventHandler into JavaPanZoomController  
17a5b8b android: key press to key event, use InputConnectionHandler  
a8f0257 android: Add callback types to Document  
3d8c72e android: Use callback types, extract string to rect conversion  
d720330 android: get LayerView from LOKitShell  
671bf23 dbaccess: use constructor feature for ORowSet  
36e4928 OFormsRootImport::CreateChildContext: survive a missing css.form.Forms  
82b0df1 SwDLL: RegisterInterfaces() is needed on Android, too  
91ca438 SwView: SelectShell() is needed on Android, too  
2a63d8d solenv: add missing component to make accelerators work on Android  
aecce90 android: integrate text selection handles from Fennec  
e1f9b53 android: Actually we don't need ViewFactory  
9288a3c Combine DBG\_UTIL and header version of Stt{End}CrsrMove  
5769734 Don't hide cursor for Android anymore - makes cursor events work  
e71b15c tdf#87098 don't adjust zoom/position for spreadsheets  
a6ae67d android: mute invalid. if not editing, don't show handle on touch  
c9893aa android: make back button work again - return false in OnKeyDown  
72e7e94 android: TextSelection improvements  
48ae33b android: gather text selection handles inside TextSelection  
e2d4cfa android: show text cursor in place suggested by invalid. event  
2f01833 android: blinking cursor  
36e701b android: fix missing drawingML preset shapes  
4b70710 lok::Document::postMouseEvent(): allow double-click  
1b15f48 LOK: add LOK\_CALLBACK\_TEXT\_SELECTION and implement it in sw  
91638b8 LOK: Return NULL when we failed to load the document.  
844ddc2 android: Update README.Android with the recent ndk-gdb info.  
d25f9a0 LOK: add LOK\_CALLBACK\_TEXT\_SELECTION\_START/END  
f36302e android: Remove unused icons.  
565a9f3 LOK: add lok::Document::setTextSelection()  
2f38b80 LOK: add LOK\_SETTEXTSELECTION\_RESET  
006c0a3 Enable LOK\_CALLBACK\_TEXT\_SELECTION\_START/END on Android  
63ae4c0 Enable mpCursorOverlay for Android too  
1eb2de9 android: add support for text selection to JNI  
c2b8fa6 android: invalidate all intersecting tiles in one LOEvent  
05b4968 android: organise imports  
71c5684 android: remove ImmutableViewportMetrics as input to endDrawing  
0e98675 android: remove tileRerender method as it is obsolete  
0ccf2c7 android: discard image buffer as soon as we upload to a texture  
a7a5ba7 android: Better check for package name when debugging.  
4928809 -Werror=unused-parameter  
8f5eff2 editeng: remove references to dead tools/string.hxx  
315e72d configure: add default value for --with-android-sdk  
4e91003 configure: add default value for --with-android-ndk  
8aa7774 SdXImpressDocument::initializeForTiledRendering: missing guard  
172eec5 android: remove LOEventFactory  
415fb79 android: remove unneeded code from LayerRenderer and LayerView  
23c4f26 android: cleanup LayerRenderer  
fd22d4b android: use mListener in LayerView for RenderControllerThread  
c207e6c android: fix "invalidate()..." exception  
b21fd93 android: cleanup more in LayerClient  
ee86272 android: check if mTileProvider is available  
72d5f0e android: hide soft keyboard  
df38cce android: Clean-up LayerClient - add @Override, rearrange, comments  
6ab77b3 android: directly call renderFrame, force render on initialization  
9236370 android: use locking to make tile redraw more predictable  
bdf8d4c android: restructure reevaluateTiles  
0622d26 android: organise imports  
bfe3a47 lok::Office::postKeyEvent: allow different char and key code  
1f20a47 android: forgot to update javadoc of JNI wrapper  
acde439 sd: initial Window::LogicInvalidate()  
484d904 sd: initial ViewShell::libreOfficeKitCallback()  
dc14b54 android: change back to LinkedBlockingQueue

# Work Report: TDF0002 – Milestone 1

a17c13e android: create SubTile right away to reduce tile change events  
402d2fb android: use Iterator for removing of tiles  
538135e android: force a screen render on redraw in LOKitThread  
b8a2e46 android: cleanup ComposedTileLayer  
f699de9 android: no need endDrawing to be synchronized  
9a51a1f android: do tile reevaluation in LOKitThread  
36e3a12 android: restructure reevaluateTiles  
6f04b04 android: organise imports  
fe4f119 android: don't hardcode touch type  
9d0950d android: make invalidations more readable - extract to own methods  
67c9a6d android: format the code on LOKitTileProvider  
0ffb877 android: bind "LongPress" to mouse double click  
eb8ed9f android: can't compare strings with == in Java  
89e9e70 android: also send key event in onKeyMultiple and onKeyPreIme  
c16de57 android: support ACTION\_MULTIPLE KeyEvent  
98115a1 android: key press/release is in KeyEvent, no need to distinct them  
2af65ff Add OutputDevice::isTiledRendering()  
8632db5 svx: paint TextEdit to the output device in case of tiled rendering  
40e3569 ImpEditEngine::UpdateViews: need to go via Invalidate() for tiled rendering  
b3fd1d5 Use SdrModel::isTiledRendering()  
d103037 android: no need for TileInvalidationCallback  
4d6eb40 android: fix LOGTAG in DisplayPortCalculator  
532e77c android: support selection invalidation, show selection handles  
ab44ada sd::ViewShell -> SdrModel for LOK callback  
49137ed android: install gdb printers  
a247d07 README.Android: document how to install the self-built .apk  
9b00d39 README.Android: document requirements for an all-symbol build  
916d40e README.Android: document how to get pretty-printers work  
77debb6 Extract LibreOfficeKitEnums.h from LibreOfficeKit.h  
  
abc3163 fix selection overlay drawing in RTL (off by one px)  
97eead6 sw: rename SwViewImp to SwViewShellImp  
a1d7f02 solenv: sfx's ScriptLibraryContainer depends on #if HAVE\_FEATURE\_SCRIPTING  
541ec41 android: LOEvents - remove unneeded, comments  
d33f321 android: compilation error in LOKitShell - SIZE\_CHANGED event  
e3cd64d android: extrac invalidations into InvalidationHandler  
8734422 android: remove DrawTimingQueue and PanningPerfAPI  
2759884 android: add comments to InvalidationHandler  
d438bb2 android: enable editing by default  
845708f Add vcl::Window::SetLastMousePos()  
b780822 Introduce SdXImpressDocument::postMouseEvent() override  
0933cad sd::ViewShell::LogicMouseDown(): typo  
80ac88c desktop: move tiled rendering debug code from sw  
60045c0 sd::Window::LogicInvalidate: add missing mm100 -> twips conversion  
cffe09 android: let LOKitThread::touch() invoke MouseButtonUp(), too  
75a186a sc: Initial support for tiled editing in Calc.  
28d01c1 sc: Make characters appear during tiled rendering as they are typed.  
f600771 ScInputHandler::ImplCreateEditEngine: don't invalidate unconditionally  
4a44051 Clean up not needed ScTabViewShell LOK methods  
c1299f0 android: change TextCursorView parent to View and draw with canvas  
5b4fc00 android: add selection support to TextCursorView{Layer}  
dc43b61 android: create InvalidationHandler outside of LOKitTileProvider  
c2ab779 android: support parsing selection rectangles, restructure code  
a854744 android: swap position of invalidateCursor and invalidateTiles  
39d4b40 android: add OverlayState to track the state of overlay  
fc4672c android: first set overlay state and send mouse button event later  
5fef3a0 android: sent -> send  
cb08fbb android: no need for MotionEvent when sending TOUCH LOEvent  
bb0892d android: add LOEvent to report handle position changes  
ed42840 android: support selection change (start,end) to TileProvider  
b2753b4 android: connect the selection change LOEvent with TileProvider

# Work Report: TDF0002 – Milestone 1

d5ad842 android: send selection/cursor change from TextSelectionHandle  
ae53e08 android: fix parsing of selection coordinates and make more robust  
b0478cb android: don't hide soft keyboard on long press  
b29e906 android: still process cursor invalidations when in selection mode  
b99fe5e android: add selection reset to TileProvider  
6c972d2 android: if in selection state, reset selection on single tap  
e85db50 android: reset selections in TextCursorView on empty invalidation  
4cbfcfe gtktiledviewer: map down/up/left/right arrow keys  
a2446dc gtktiledviewer: map shift key  
135bc56 libreofficekit: pretty-print LibreOfficeKitCallbackType  
f52920a SwShellTableCrsr::FillRects: ignore visual area for tiled rendering  
ee8ee08 SwSelPaintRects: add FillStartEnd() interface  
645bebd SwSelPaintRects::FillStartEnd(): missing const  
6650fe9 LOK: move Office::postKeyEvent to Document  
8c086b9 SwEditWin::SetCursorLogicPosition: use SwCrsrShell::getShellCrsr

# Work Report: TDF0002 – Milestone 1

## Appendix B: Commits Related to Android work

These commits are included for interest, as part of Collabora's ongoing LibreOffice on Android investment, and are generally not included in Deliverables, while being potentially relevant to the work and of interest for TDF.

### General Android bug-fixing

A number of issues reported by users based on reported crashes, failure to load files etc. were fixed alongside the Deliverables:

*Commit ID*      *Commit message*

```
6debb04 android: show vendor in LOAbout
aff19a6 android: add dummy resource to get the correct package name
a468472 android: revert the app package name & relative activity name
4bd3154 android: org.libreoffice -> org.example as default package name
d38b661 android: make the about dialog scrollable
609367d android: Set the package name correctly.
722d4f0 android: Fix release build via top level 'make'.
2b67566 android: Set the debug / release according to the configure setting.
2d584cb android: Bind the release/debug setting to --enable-release-build.
ae924ed android: Rename 'make sign' to 'make release-apk'.
b25dd85 android: In Impress, switch to the 'Normal' (slides) view.
9b482b3 android: Initialize for tiled rendering earlier.
0059a21 android: When loading fails, make sure we don't crash the next time.
24b2815 android: allow only "file" and "content" URI scheme
768dea1 android: copy document to temp file when using content scheme
6cdd02a android: protect against a crash with an empty sdcard.
a246ad3 android: Avoid big amount of space below the icons.
```

### General Interoperability component inclusion

This involves running our bugzilla documents through a viewer-like object file under Linux, to detect where we are failing to load documents due to missing components. Then fixing these call-sites, and/or improving the way that components are split to provide better fidelity.

*Commit ID*      *Commit message*

```
e69fb5a native-code.py: We need these services everywhere anyway.
98279ce native-code.py: Add a mode that allows filtering out services.rdb.
5d86b4b native-code.py: Sort the constructors.
68210e1 native-code.py: Share more code around the factories too.
2c27ad3 native-code.py: When filtering the services.rdb, keep also the
factories.
73076f5 forms: use constructor feature for OFormsCollection
a11c71f forms: use constructor feature for ODatabaseForm
c84cc4f native-code.py: Add libi18nsearchlo.a.
8e58af2 Add slideshow-related services.
1eb4114 android: Always log the SAL_WARN's.
9fc47fb native-code.py: Add libxolo.a.
17be6da native-code.py: Add libfrmlo.a + some constructors around controls.
146461e Add an overlooked constructor to chartcore.component.
e24ca72 native-code.py: Don't add the entire libchartcorelo.a.
e27387a android: Set a more verbose SAL_LOG.
17da989 android: Even --enable-debug needs org.libreoffice for debugging.
8ca6490 native-code.py: Add libevtattlo.a + some forms related services.
3d9b364 It is not necessary to be _that_ verbose.
baa5459 android: Remove duplicated icons.
2e70fa6 android: optimize the resources.
1e4d83a native-code.py: Add com_sun_star_comp_dba_ORowSet_get_implementation.
917bd23 Use constructors for services from chartcore.component.
4ed53df android: No need to include the entire libfrmlo.a, add just the used
services.
99a0942 filter: use constructor for filter module
```

# Work Report: TDF0002 – Milestone 1

b8a5e0c writerfilter: use constructor for writerfilter module  
712fc6e native-code.py: Writerfilter implementation is now converted to constructors.  
e1879e6 stocservices.component: The rest of the services + cleanup.  
1fc8f8a native-code.py: stoc implementation is now converted to constructors.  
735cc12 native-code.py: Missing services after the stocservices conversion.  
232846d native-code.py: Add more missing chart services.  
bc81d76 native-code.py: Add more forms constructors.  
cd23128 sw: convert import & export components to constructor usage  
80bcdb1 sw: convert SwWebDocument & SwGlobalDocument components to use constructor  
94fc8ec sw: convert swd.component to constructor usage  
a45a550 basic: convert sb.component to constructor usage  
7639858 native-code.py: Some services were converted to using constructors.

## GtkTiledRenderer development test-bed

*Commit ID      Commit message*

d9997df gtktilviewer: register a LOK callback and re-render the document ...  
057fb14 LibreOfficeKitGtk: lok\_docview is never read  
92e3508 gtktilviewer: react to key presses  
8d8729c gtktilviewer: react to key releases  
160e3c6 gtktilviewer: invoke lok::Office::postKeyEvent() on key press / release  
9563293 gtktilviewer: invoke paintTile() on the main thread from the callback  
a84157c gtktilviewer: handle more key values  
9adaaa3 gtktilviewer: missing initializeForRendering()  
fe3ab84 gtktilviewer: start in viewer mode, switch to edit mode by mouse click  
1e3b09d lokdocview: clean up renderDocument()  
58959bd lokdocview: stop rendering a single huge tile  
c4cdb69 lokdocview: adapt comment to match reality  
7e01796 lokdocview: add LOKDocViewCallbackData  
86fba50 lokdocview: add support for partial rendering  
c3f1a33 lokdocview: reduce scope  
90ffb3b gtktilviewer: show error message when lok\_docview\_open\_document() fails  
8807da1 gtktilviewer: call postMouseEvent()  
a6276c5 lokdocview: send mouse events in doc model coordinates  
883ae86 fix for missing g\_info  
d6587d7 gtktilviewer: fix for missing g\_info()  
64235bc lokdocview: fix for missing gtk\_table\_get\_size()  
31ce5f4 lokdocview: initial overlay on top of the tiles  
4c51ea9 lokdocview: let the cursor blink  
5d483a6 lokdocview: fix memory leak in lcl\_payloadToRectangle()  
c0a185e lokdocview: implement selection overlay using LOK\_CALLBACK\_TEXT\_SELECTION  
e8fe158 lokdocview: fix Linux baseline build  
5dee910 lokdocview: when have a cursor but no selection, show the middle handle  
e07d9c0 lokdocview: draw handles at selection start/end  
13de1b8 lokdocview: avoid loading the handle bitmaps again and again  
0135d5d lokdocview: allow dragging the middle handle  
be77714 lokdocview: allow dragging the selection end handle  
5e5649a lokdocview: allow dragging the selection start handle, too  
f26dd84 lokdocview: reset start/end or middle handle depending on if we have selection  
54b231e gtktilviewer: map css::awt::Key::ESCAPE  
74a9840 lokdocview: scale selection rendering / cursor handling with zoom.  
e900aae gtktilviewer: map GDK\_Tab to css::awt::Key::TAB