

Work Report: TDF0002 – Milestone 2

Development of Basic Editing Capabilities for LibreOffice for Android – Milestone 2

This project targets implementing basic editing in the LibreOffice for Android code. The second milestone demonstrates that the functionality required for the project is complete.

Table of Contents

Development of Basic Editing Capabilities for LibreOffice for Android – Milestone 2.....	1
Overview of the Task.....	2
Addressing Comments from M1 Acceptance.....	2
Documentation concerns.....	2
New bullet not appearing.....	2
Tap-to-edit not working in Calc.....	3
Selection and cursor at unexpected location in Impress.....	3
2.1.6 (10.1.6) Performance Considerations, Support for OpenGL.....	4
2.1.7 (10.1.7) Rendering of Selections on Overlays.....	5
Writer selections.....	5
Impress/Draw selections.....	5
Calc selections.....	6
2.1.8 (10.1.8) Routing of Application Information from the Program Module’s Core.....	7
The other way around, routing the information from Android to the program module's core: manipulating the selections.....	8
2.1.9 (10.1.9) Text, Tables for Writer and Impress, Tables for Calc, Pictures and Shapes.....	9
Development and Documentation.....	10
Community Engagement / Public Promotion.....	10
Conclusion.....	10
Appendix A: Commits Related to Deliverables.....	11
Appendix B: Calc Tiled Rendering Implementation.....	15
Appendix C: Commits Related to Android Work.....	15
Appendix D: General Android and Other Bug-fixing and Cleanups.....	15
Appendix E: GtkTiledRenderer Development Test-bed.....	16

Work Report: TDF0002 – Milestone 2

Overview of the Task

The first Milestone was to deliver five of the relevant 2.1 items from the contract – including relevant items from 2.2 and 2.3. 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 for Milestone 1 were: 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).

Milestone 2 completes the remaining items.

Addressing Comments from M1 Acceptance

Documentation concerns

We have improved documentation at several levels, as planned, since M1:

1. The Android Java UI code (each class and method added during this work) now has javadoc comments.
2. Not only the editing-related, but all other LibreOfficeKit API methods have doxygen comments.
3. Architecture of the Android part is now documented in android/README.
4. As before, architecture of LibreOfficeKit is documented in libreofficekit/README, and each newly introduced C++ class and method have doxygen comments.

In addition to documenting every method and class that we *added*, we have gone further to document some methods that were simply altered or pre-date the Deliverables. Reviewers should be aware that this extra work was done on a best-effort basis and that not all pre-existing methods (particularly trivial ones) are commented.

Additionally, we also improved the SdrObject/drawinglayer related documentation in svx/README.

New bullet not appearing

There was a problem with keyboard input handling: every other call between Android and the core calls was executed on a thread which takes the solar mutex. In contrast, keyboard events were posted to the main loop of the soffice thread. The result of that was that in case of many Android to core calls, the main loop executed its scheduled events with in a delayed manner. We have fixed that, so now every action is executed on a thread. Our testing shows that this fixes the effect that some users experienced missing / delayed processing of keyboard input.

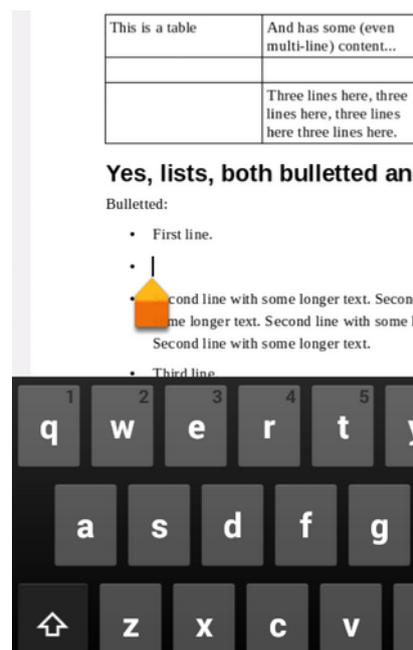


Figure 1: Bullet inserted to the reference ODT document

Work Report: TDF0002 – Milestone 2

2.1.6 (10.1.6) Performance Considerations, Support for OpenGL

The work as it is uses OpenGL on the device in the sense that the bitmaps (tiles) provided by the LibreOfficeKit are up-loaded as OpenGL textures to the GPU and handled there to provide a composited view at the end, which gives nice panning and zooming experience.

We evaluated the state of porting VCL to OpenGL. At the time of outlining the contract, there was hope that OpenGL will be used by default on the desktop by now and ripe for re-use on Android. That did not happen yet: OpenGL on the desktop is still hurt by some lifecycle problems, and a few rendering artifacts have to be fixed before that can happen. This means we continue to use the basgfx CPU rendering.

We have evaluated the performance, and after some optimization found it to be acceptable: There is no need to provide a cumbersome non-WYSIWYG overlay for editing. For image and shape resizing, instant re-laying out would be expensive and problematic from a UI perspective, so we show a preview of the bounding box, and resize and re-layout only after the final bounding box position and size is known.

We found two scenarios where performance was really poor:

1. On loading Impress documents, not only the main slide view, but also the slide sorter was rendered (invisibly) in the background. The Android-specific mobile-config.py script, that adapts the desktop default config to Android requirements now disables the slide pane by default.
2. We had a look at the amount of invalidations after different actions, and found that initially Impress re-drew the current slide six times on each touch event: which made editing slower than necessary. This was a tiled-rendering-specific quirk fixed by not changing map-mode too often. After the fixes, we found the Impress text editing performance acceptable.

Another performance-related problem was that in some cases we were providing the tiles too quickly: LibreOfficeKit did not wait for the Writer layout to finish, so the result was sometimes incomplete. The presence of these partially rendered white tiles caused significant ugliness when zooming in some cases. . This is now fixed, no tile is handed out till the relevant part of the layout is finished. This means that we can still show the first page of a document in advance of subsequent pages' layout calculation, and still be correct.



Figure 4: Tile showing the first page, before.

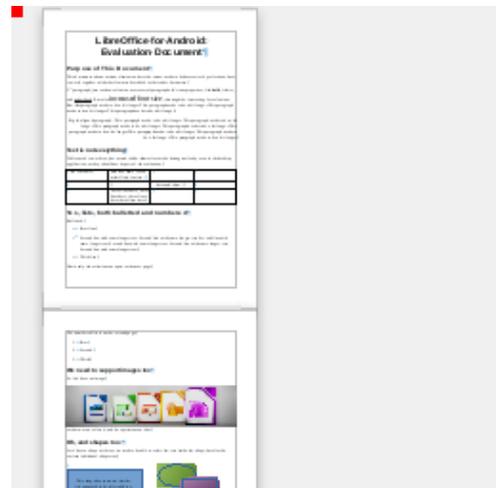


Figure 5: Tile showing the first page, after.

Performance could be further improved in the future using a profiler on Android to see when the time is spent on the CPU and on the GPU. Unfortunately, our partner who had volunteered to do performance profiling, and for whom we created a ~2.4Gb binary APK including full debugging symbols – was not able to make their profiler work against LibreOffice on Android. The absence of good profiling data (and debugging tools) makes native code development on Android unusually unpleasant and problematic.

Work Report: TDF0002 – Milestone 2

2.1.7 (10.1.7) Rendering of Selections on Overlays

We introduced a layer for drawing the selections as a set of alpha transparent rectangles at locations provided by LibreOfficeKit. Essentially, on Android we handle two types of selections: text selection (used also for tables) and graphics selections. These have to be painted differently, and also differ in the behavior.

In the further text, please see details of Writer, Calc, and Impress selections.

Writer selections

We implemented the following selection types:

1. Text selection
2. Table selection
3. Graphic selection (both Writer images and SdrObject/drawinglayer shapes)
4. Graphic text selection in drawinglayer shapes

In all cases we instrumented core to emit LibreOfficeKit callback events containing the correct coordinates of the selections so that all special cases are handled. Briefly considering a text selection starting and ending at different lines or even different pages, reveals that the list of selection rectangles is not necessarily trivial.

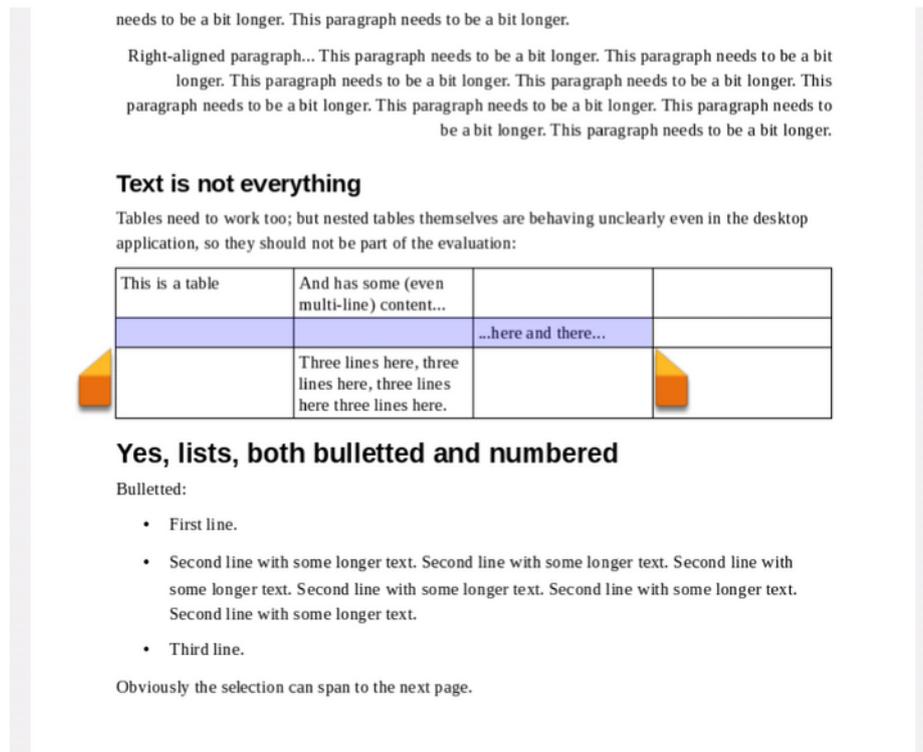


Figure 6: Table selection in Writer.

Impress/Draw selections

We implemented the following selection types:

1. Image and shape selection
2. Shape text selection
3. Impress table selection

Work Report: TDF0002 – Milestone 2



Figure 7: Image selection in Draw.

Calc selections

We implemented the following selection types:

1. Selection of cell text
2. Selection of cell ranges
3. Graphic selection (images and shapes)
4. Graphic text selection

Please note that in some cases, the selection of cell ranges looks slightly misplaced. This is a pre-existing problem tightly coupled with how the drawing works in Calc – it is all happening in pixels, and consequently all kind of rounding errors appear. On the other hand, the selections are defined in 100ths of millimeter – so technically the selections are at the 100% correct positions, but the underlying cells are slightly off.

Rewriting the Calc rendering is out of scope, as that is not related to selections, though we still want to address the problem later via a GSoC task mentored by our staff.

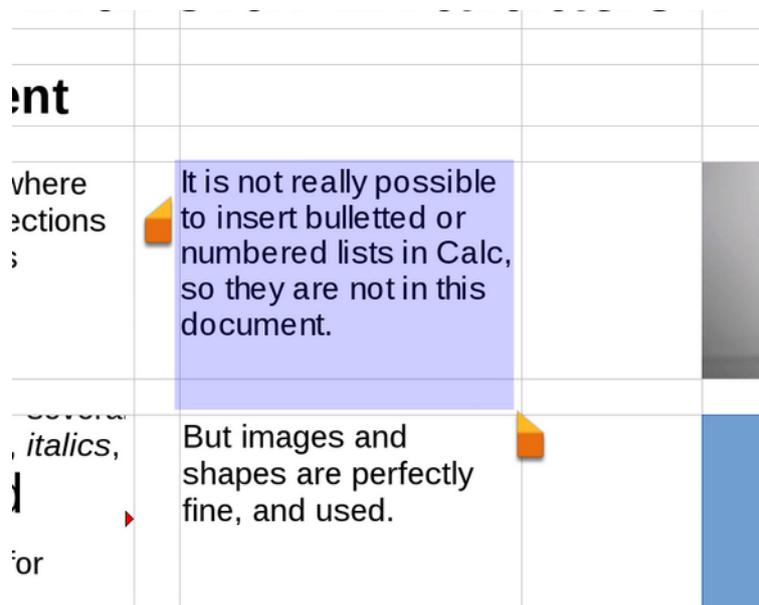


Figure 8: Multiple cells selected in Calc.

Work Report: TDF0002 – Milestone 2

2.1.8 (10.1.8) Routing of Application Information from the Program Module's Core

This was a considerably large amount of work, as we had to implement two features here: exposing the cursor state via a LibreOfficeKit callback, and exposing the same for selections.

While the view of the Android application is completely de-coupled from any live concept of a view in the LibreOfficeKit core, selections and cursor position are owned and managed by the core. When rendering, the Android state and the core state (ie. what is visible to the user) are not in sync: it is necessary to eg. provide tiles at various resolutions, or it may be necessary to re-draw the whole output when scrolling – when using tiles, the Android app can manage this itself much more efficiently. The selections and cursor handling is on the other hand more complex than rendering, and closer to the document model, which makes it necessary to let core manage the state.

The cursor has two important properties: whether it's visible (exposed via `LOK_CALLBACK_CURSOR_VISIBLE`), with position / size (exposed via `LOK_CALLBACK_INVALIDATE_VISIBLE_CURSOR`). The cursor is not visible when not editing (only viewing) or when there is a graphic selection. In core, there can be many `vcl::Cursor` instances, and then it is ensured that only one of these cursors are visible at the same time, that's how the user sees the position of "the cursor", be it a shape text (editeng cursor) or a Writer one (sw cursor). The same is reflected in the LibreOfficeKit API: when one cursor is visible, a LibreOfficeKit callback event is emitted to say that "*the cursor is visible*". This way the Android UI doesn't have to deal with the complexity of multiple cursors. As for the position/size: The text cursor is described as a rectangle, although the width of the rectangle is 0, and it is up to the Android UI to determine what is a natural minimal width for the cursor caret.

Regarding selections, we implemented the text and the graphic selection. Text selection is a list of rectangles (exposed via `LOK_CALLBACK_TEXT_SELECTION`), with an explicit start (via `LOK_CALLBACK_TEXT_SELECTION_START`) and end rectangle (via `LOK_CALLBACK_TEXT_SELECTION_END`). This is necessary because the Android UI could not correctly find out the start and end of a selection, e.g. an A1:B2 table selection has the start rectangle based on the A1 cell's height, but that's unknown to the Android UI, so it could not show the selection handles at the right position. Graphic selection is simpler, there we only need to provide a rectangle (`LOK_CALLBACK_GRAPHIC_SELECTION`) for the image or shape.

Given that in LibreOffice these selections are implemented differently, we had to instrument multiple places to emit the same events. For example, `LOK_CALLBACK_TEXT_SELECTION_START`, the selection start event is emitted at the following places:

1. editeng, used for text in shapes (Writer/Impress/Draw/Calc) and for Calc cell text
2. sc, used for Calc cell selection
3. svx, used for Impress table selection
4. sw, used for Writer text and table selection

All these places implement the same LibreOfficeKit API, describing the same feature using a unified language: working in logical document-model coordinates, in TWIPs.

Work Report: TDF0002 – Milestone 2

Table in Impress

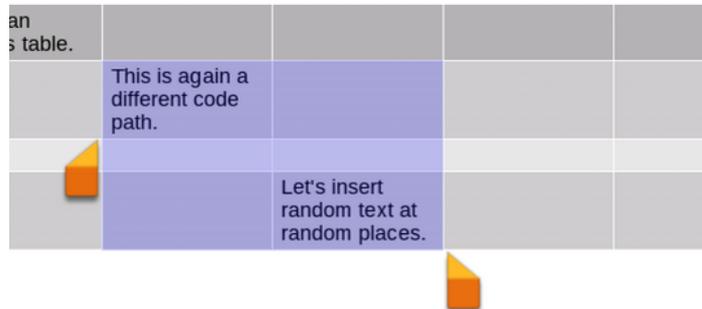


Figure 9: Table selection in Impress.

The other way around, routing the information from Android to the program module's core: manipulating the selections

All the above is exposed as text and graphic selections at a LibreOfficeKit level to the Android UI. Table selections are also exposed as text selections, and not as a special kind of selection, which gives the opportunity to seamlessly change the text selection into selection of table cells. Still, this way of selecting cells provides the handles as specified in the contract.

We implemented the following ways to manipulate the selections:

1. For text selections, the start and end of the selection has a handle, which can be dragged to shrink or expand the selection. This works both for text (more/less characters) and tables (more/less cells or rows). Once the text selection becomes a table selection, the user has to cancel the selection if he/she wants to return to selecting parts of text again.
2. For graphic selections, the selection has 8 handles to extend or shrink the size of the image/shape. Moving is possible by dragging the shape itself, and not the selection handles.

Selections can be created the following ways:

1. Long push on a word (in Writer's paragraph, Calc's cell or inside a shape text) creates a text selection.
2. Long push on an empty Writer, Impress/Draw or Calc cell creates a cell selection.
3. Dragging the text selection handles of a Writer, Impress/Draw or Calc cell outside of that cell creates a cell selection.
4. Tapping on the border of a Writer Image, or inside SdrObject/drawinglayer image or shape (if the shape does not have text) creates a graphic selection.

Work Report: TDF0002 – Milestone 2

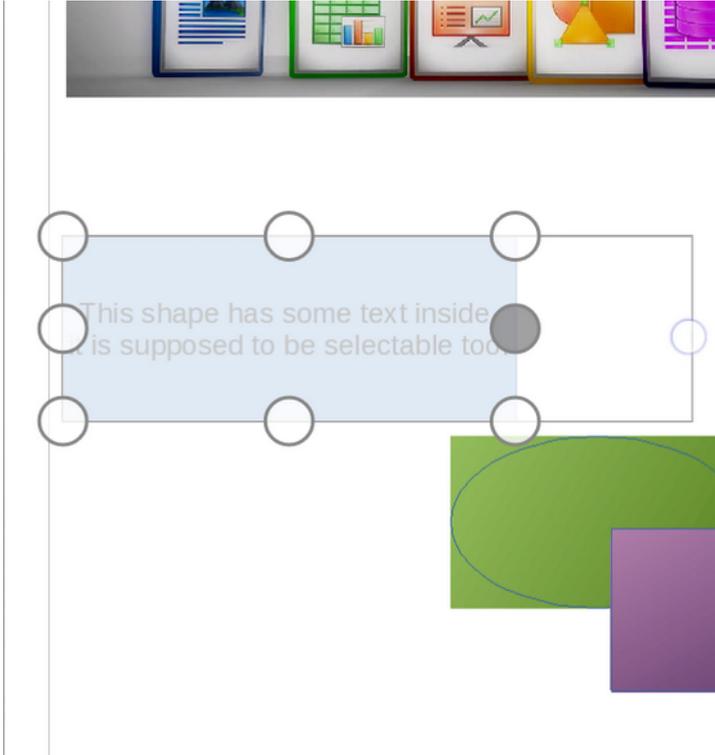


Figure 10: Preview of shape resize in Draw.

Work Report: TDF0002 – Milestone 2

2.1.9 (10.1.9) Text, Tables for Writer and Impress, Tables for Calc, Pictures and Shapes

This task was a logical conclusion of the above mentioned tasks – we connected the rendering and events together, and extended the right code points in the LibreOffice core.

Here is a list of places where we emit selection events:

1. Writer text selection: provided by the SwSelPaintRects and SwShellCrsr classes, responsible for selections of characters, words, paragraphs, lists, text in tables.
2. Writer table selection: provided by the SwShellTableCrsr class, responsible for selection of Writer tables (selection of entire cells or cell ranges).
3. EditEngine text selection: provided by the ImpEditView class, responsible for the text in drawings, images, and Writer shapes, and for the Impress/Draw text and Calc text.
4. Graphic selection: provided by the SdrMarkView class, responsible for the bounding box of the selected (“marked”) shapes and images.
5. Calc cell selection: provided by the ScGridWindow class, responsible for Calc cells and cell ranges.
6. Impress/Draw table selection: provided by the sdr::table::SvxTableController class, responsible for selection of entire cells of cell ranges.

Second line with some longer text. Second line with some long
some longer text. Second line with some longer text. Second li
Second line with some longer text.

Third line.

ously the selection can span to the next page.



Numbered list is on the second page:

- . First.
- . Second.
- . Third!



need to support images too

It's have an image:



Figure 11: Multi-page selection in Writer.

Work Report: TDF0002 – Milestone 2

Development and Documentation

All implemented features described previously are committed to the main LibreOffice git repository, to the master branch.

We communicated our progress with the project since M1 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 two new entries in addition to the three enumerated in the M1 report:

1. <http://vmiklos.hu/blog/tiled-editing-part-4.html>, titled “Android editing: from selections to graphic handling”
2. <http://vmiklos.hu/blog/android-editing-part-5.html>, titled “Android editing: from graphic handling to formatting”

Regarding architectural documentation, android/README describes the structure of the Android UI, while libreofficekit/README describes how to use the LibreOfficeKit API, with focus on the editing capability added during this project.

Additionally, every C++ class and method added during this project is documented using doxygen. The resulting documentation is regularly published at <http://docs.libreoffice.org/>.

We also presented the project at FOSDEM2015.

An archive of all slides and blog entries can be found at <http://people.freedesktop.org/~vmiklos/2015/android-editing-archive.tar.xz>.

Community Engagement / Public Promotion

We produced five 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 students in a Google Summer of Code role in turning gtktilviewer into something more powerful for use in GNOME Documents and to improve the Android app further during the summer, though the selection process will show if these are strong enough candidates.

Conclusion

We believe we have delivered all of the functionality required by the contract on time, and in many cases we have gone even further: especially in the area of Calc, where the tiled rendering was in an inappropriate state before starting this work. We also hope that our write-up highlights the main points of the work, that you are happy with the achievements, and we look forward to including you in a successful demonstration of the acceptance tests on Tuesday 14th April.

As with all Engineering tasks, this was a team effort. Naturally without the huge investment from the LibreOffice developer community, and TDF's donors it wouldn't be possible to have done this. Having said that, it is good to recognize the outstanding work done by the Collabora team members:

- **Miklos Vajna** designed and implemented the editing parts in the LibreOffice core, LibreOfficeKit, Writer and Impress. He has also done most of the changes in the gtktilviewer, our extra test application.
- **Tomaz Vajngerl** worked on the Android parts, and implemented the graphics and behavior changes that were needed to plug the new LibreOfficeKit functionality in with help from **Siqi Liu**.
- **Henry Castro** and **Kohei Yoshida** reworked the Calc tiled rendering, and implemented the editing parts in Calc with much help from **Jan Holesovsky** who also oversaw the project.

A final issue remains around test-ability of the editing functionality. While APKs are available from the tinderbox, there is no direct channel for users & testers to play with this on Android. Collabora currently 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 perhaps some suitable gateway dialog crediting and soliciting donations for TDF in the interim 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.

Work Report: TDF0002 – Milestone 2

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>
94e5223	android: account for handle repositioning, limit update to 50ms
4cb40de	SwShellTableCrsr::FillStartEnd: make sure that the rectangle's width is not 0
b2b3a18	SwEditWin: when tiled rendering, let double click select an empty cell
2c9ca62	SwCursor: when tiled rendering also select word before the point
c86b106	android: fix mHandlerType initialization
b41b968	android: fix long press not selecting an empty cell
b99e0dd	desktop: avoid restart when running in a LOK unit test
3687ae6	CppunitTest_libreofficekit_tiledrendering: fix up and enable this again
cc7d10c	SwEditWin: disable map mode in LogicMouseButtonUp/Down
5fb18e1	LOK: add LOK_CALLBACK_CURSOR_VISIBLE
7e189aa	Use SwCrsrShell::SelTblBox()
4c7e490	SwViewShell: store tiled rendering state in SdrModel
706691b	android: rename text selection callback messages
7373248	android: add and handle show/hide cursor message from LOK
1b36343	android: also hide soft keyboard when hiding cursor
0c4b539	android: initialize TextCursorView with cursor invisible
6d0d103	sw: when tiled rendering, disable map mode of SwEditWin early
233fdd9	LOK: add LOK_CALLBACK_GRAPHIC_SELECTION API and implement it in svx
907fdbf	add getPartsCount() to TileProvider
a5b384c	add license header
5aa19be	sw: move LOK_CALLBACK_CURSOR_VISIBLE event from SwVisCrsr to SwCrsrShell
7c1a228	Add sdr::overlay::OverlayManager::getModel()
658f762	sdr::overlay::OverlayObjectList: when tiled rendering, don't work with pixels
3650a9f	sc tiled mouse events: Proof-of-concept.
58c89e4	android: support to make selections visible/invisible
d5c32b4	android: minor code clean-up in TextSelectionHandle
b792f60	android: handle callbacks directly in InvalidationHandler
1a8b898	android: do all overlay state tracking in InvalidationHandler
a54fe6d	android: fix rebase import problem
049aa53	android: add CALLBACK_GRAPHIC_SELECTION to LOK facade
8c5a900	android: don't hide CursorView by default
8f73bb4	Whitespace in a recent commit.
2e29397	add option to disable animations, use in dashed border overlay
80c90f5	android: add graphic selection
bdd8a68	android: make & build appcompat-v7 library, up target to v21
4cf34bf	android: actually use the new toolbar in the application
bd03308	use toUtf8 instead for string conversion
1fd5450	sc tiled mouse events: Fix the click coordinates in non-100% zooms.
f030a7f	Consistency ;-)
7f4e5c82	drawinglayer::processor2d::HitTestProcessor2D: ignore pixel size of ...
a14d292	android: wrong parent in theme definition
003a62f	android: add graphic selection to TextCursorLayer{View}
b6df97c	android: simplify state tracking, make graphic selection work
1771f21	android: use LOK_SETTEXTSELECTION_RESET as it is meant to be used
c08069f	android: clicking on toolbar icon exits edit mode
65726aa	LOK: add lok::Document::setGraphicSelection() API and implement it for Writer
0262693	lok::Document: add resetSelection()
34f3f24	make format consistent with the rest of the file...
2c0498d	android: add some material design icons
598ba92	android: add toolbar controller
bcc3163	android: add bold selection to toolbar (experimental)
bd3771e	android: add to JNI - setGraphicSelection, resetSelection
ea60724	Small cleanup - no need to have 2 OpenURL's.
73ed546	SdrMarkView::SetMarkHandles: when tiled rendering, show non-frame

Work Report: TDF0002 – Milestone 2

handles, too
a1ae91f Add testcase for SwXTextDocument::resetSelection().
ab27a77 Add SwXTextDocument::setGraphicSelection() testcase
4a8afa0 Add ImpEditView::libreOfficeKitCallback()
a250e9d Factor out the .uno: command dispatching to a separate function / file.
00b758b LOK: Implement posting of .uno: commands.
2ba8f4d editeng tiled rendering: emit callbacks necessary to have a blinking cursor
5df6cd1 android: Make the Bold button actually work - switches to typing in bold.
39bf824 LOK: extract the callback typedef to a separate LibreOfficeKitTypes.h header
6e8da9b libreofficekit: add overview of tiled editing to README
fa53289 SwXTextDocument: add missing @see for 3 vcl::ITiledRenderable methods
6e90795 android: hide soft keyboard when scrolling
8654e16 android: also hide keyboard on graphic selection
30d83b0 Add SwXTextDocument::setTextSelection() testcase
13a06aa SwSelPaintRects::Show: silence LOK callback till draw edit is active
af379e9 ImpEditView::DrawSelection: emit TEXT_SELECTION_LOK callback
2479fae android: add toolbar button to force-show the soft keyboard
81f0c28 lok_docview_init: ignore the second, not the first param
e6fb43c CppunitTest_sw_tiledrendering: disable part of testSetGraphicSelection on OS X
43a503c android: add missing license headers and modelines
8925424 ImpEditView::DrawSelection: emit TEXT_SELECTION_START/END callbacks
2a081a0 Rectangle::toString(): output position, then size
2f31ac7 editeng tiled rendering: allow resetting selection with a single click
a212fc2 Add SwXTextDocument::postMouseEvent() testcase.
f0f0ce2 android: add comment to many classes and methods
7ba960e android: clean up no longer needed com.polites library in LOAndroid3
c4f3d69 editeng tiled rendering: implement drag of the middle selection handle
82bc350 editeng tiled rendering: implement drag of start/end selection handle
b3ef2a7 Add SwXTextDocument::registerCallback() testcase.
772e5c9 sw tiled rendering: let single click start shape text editing
93fbb73 android: TileProvider is unused in InvalidationHandler
b053778 android: set graphic selection via TileProvider
86548b6 android: move graphic selection, use a touch event to transmit
907583d android: change cursor blink time to a constant static field
e48d983 android: draw also the center handles for graphic selection
7f4ac90 android: move handle code to DrawElementHandle
892d681 android: Hit test for handles, show resize rect
d9bc184 setGraphicSelection should use LOK_SETGRAPHICSELECTION_*
61ebb1b sc tiled editing: Use the correct device for PixelToLogic.
83f2516 sc tiled editing: Clear the right area behind the editeng.
2bf1955 sd tiled rendering: draw documents have no slides view
6d9610c android: a bit smaller hit area for selection handles
718143b svx: if the handle bitmap is empty - create a replacement bitmap
1497924 android: add a check for the installed memory in the machine
c5c2d5242 gtktilviewer: remove unimplemented gtk-zoom-fit button
658534d SdrMarkView tiled rendering: suppress handles during text edit
b0cb454 android: add debugrun Makefile target
2b5cf23 android: when in selection, avoid showing the middle handle
a20167b fold README.Android into android/README
f9c9f57 OutputDevice::LogicInvalidate: clean up sc/sd duplication
97ea96f ImpEditView::ShowCursor: don't assume map mode is twips
35cf00e OutputDevice::LogicInvalidate: take a Rectangle
7b8a598 sd: when tiled rendering, disable map mode early
19c59a3 android: remove import android.util.Log from InvalidationHandler
205b695 android: extract drawing of selection into its own class
a9fd1af android: reorganize (overlay, canvas packages), move classes
913b3a6 Experimental: draw handles instead of getting them from bitmap
c1434d6 sdr::table::SvxTableController::onMouseMove: remove dead code
79dac94 android: add doc on how to set up jdb

Work Report: TDF0002 – Milestone 2

25990de vcl tiled rendering: fix invalidation callback in Impress
a074ebf editeng tiled rendering: fix selection overlay in Impress
0cc8eb1 svx, sw: can avoid vcl/ITiledRenderable.hxx include here
c040ca8 vcl::ITiledRenderable: can mark these methods as pure virtual
fc3896a SdXImpressDocument: implement setTextSelection()
e23af8b svx tiled rendering: fix Impress graphic selection
c14a266 sc tiled editing: Setup the Calc's editeng for tiled editing.
efbf462 sc tiled editing: Implement moving of the selection handles.
78dee9c SdrObject::dumpAsXml: show bounding rectangle
e7cf3c0 SdrMarkView::SetMarkHandles: perform the mm100 -> twip conversion on a copy
9eb26a9 svx tiled rendering: grow hittest size when map mode is in mm100
0d6a07f sd tiled rendering: increase hit testing tolerance
3fdb48d SdXImpressDocument: implement setGraphicSelection()
0d61a74 vcl::ITiledRenderable: setTextSelection() can be pure-virtual now
befa877 android: fix wrong TextCursorView class name, format xml
5d31f2d No need to create ScAddress here.
9294dc3 Minor tweak of the LOK_CALLBACK_TEXT_SELECTION_START / _END docu.
7609c0b sc tiled editing: Implement cell selections.
5e43b9a SdrMarkView tiled rendering: fix unexpected empty graphic selection events
dd3eb52 These checks are cheap, simplify.
a7e75be sc tiled editing: Small refactor, preparing for moving cell selections.
7fa8f72 sc tiled editing: Allow resizing the cell selection using the handles.
0bc8e67 sc tiled editing: Make the selections work with zoom too.
aa5a78c android: move handles into GraphicSelectionCanvasElement
2f6c4d9 android: increase handle hit area
ba93e70 android: fix handle and selection behaviour, clean-up
b1def00 android: introduce CanvasElement interface
16d8839 android: rename graphic selection and handle
3aed864 Make SetCellSelection work in pixels, and change the name accordingly.
f951ab5 sc: No need for a special LogicMouseButtonUp/Down.
24587ec sc tiled editing: Implement scaling of drawings and bitmaps in Calc.
25f0d57 android: add contains (hit test) to CanvasElement interface
4415ecb android: properly merge README with new content
00094f9 sc tiled editing: Implement resetSelection().
741599f SdrMarkView tiled rendering: partially disable SdrTableObj graphic selection
62d03f6 SdrTableObj tiled rendering: implement selection callbacks
84793d1 SdXImpressDocument: implement resetSelection()
54510b3 svx tiled rendering: double-click in empty table cell selects it
a518d5a android: open csv files in Calc
eba73d9 android: comment selection related code, rearrange and clean-up
99394c8 Add SvxTableController::setCursorLogicPosition()
9987747 sc tiled editing: Improve the deselection.
7fbfd04 sc tiled editing: Hide the cell selection when we select graphic.
f3915ff vcl: fix GCC build
32cfef5 LOK: document remaining Document member functions
046be64 sd tiled rendering: support turning an editeng selection into a table one
2aa1435 SvxTableController::setCursorLogicPosition: fix unexpected graphic selection
812970c android: fix sending graphic selection type to LOKitThread
d7716c9 android: enable AA for selections and handles, tune colors & alpha
9dfaffa android: minor comments change
f15a6a0 android: fix 'make install' in LOAndroid3
f8bd968 Add testcase for sd::Window::LogicInvalidate().
8eeb6ef sd::ViewShell::LogicMouseButtonUp: typo
656d3f9 android: register TextSelection & TextCursorLayer once at creation
626e1a2 android: set layerView to TextCursorView{Layer} when available
6d50c72 android: single press on a selected shape triggers text edit
41b3ed6 android: convertPosition -> convertToScreen
d009630 android: isVisible, setVisible for CanvasElement + common impl.

Work Report: TDF0002 – Milestone 2

7a53938 android: use isVisible setVisible for GraphicSelection
8f6b3d6 android: use onDraw for element drawing in CanvasElement interface
f8f30f6 Change the Logic in SetGraphicLogicPosition()
6448e86 SdrPaintView tiled rendering: ignore visual area
2e84d5e android: Don't show canvas elements by default
492e52c android: add assign which copies rect coordinates from source
bf995db android: add onHitTest for implementations, add impl. interface
6d59210 android: replace TextSelection{Handle} with SelectionHandles
9a4d0bb Prevent showing the dialog (and crashing) when tiled rendering.
01a1cf2 android: propagate visibility to graphic selection handles
2502121 android: extract BitmapHandle from SelectionHandle
21bc58d Add SdXImpressDocument::postMouseEvent() testcase.
2f230db sc tiled editing: Show the cell cursor as a rect, not as a frame.
4d9532d sc tiled editing: Don't show the cell selection when editing text in shape.
d351344 Kill some whitespace found during debugging.
06a1cbf sc tiled editing: Make the editing of text in shapes in Calc work.
428c3a7 sc tiled editing: Invalidate graphic selection.
cf405bb vcl tiled rendering: fix Writer/Impress invalidation rectangle
a72a026 sc tiled editing: Moving handles in text in drawing shapes in Calc.
6218409 android: update README
03ee2793 SwSelPaintRects: avoid rendering the overlay on Android
a1605d6 sc tiled editing: Use the LogicMapMode even for mouse positions.
7091be1 Update comment.
1a76acd sc tiled editing: Set the 100% zoom in the selection-related methods too.
c768fc1 sd::ViewShell::SetGraphicMm100Position: can avoid FuPoor::DRGLOG here
a9373a3 vcl tiled rendering: don't do anything with disabled mapmode and mm100 mapunit
1c4202d android: remove selection handles from xml definition
dacf5c4 android: don't reset position at showCursor
3dbb8cc android: don't update position, if it is the same
2bbf630 android: don't modify if the value is same - all in TextCursorView
ae4f4b7 android: remove unneeded layerView stuff
7297feb6 LOK: reimplement lok::Document::postKeyEvent()
d4a39c1 android: Insure -> Ensure
a2222ba Add SdXImpressDocument::setTextSelection() testcase.
f135e08 Added some ascii art for the SdrObject and SdrView.
a595cf0 More additions about SdrView / SdrModel.
026004e sc tiled editing: Don't show cell selection when dragging shapes.
ec86462 sc tiled editing: This correction is not needed, pass the exact values.
3f82a2e android: remove dead GSoC-2012-eclipse-workspace
7bc83da android: update startup details of README
ee1e091 sc tiled editing: Changed behaviour double click to click.
ac662cc sc tiled editing: Correct position of the selection in far positions.
577eb509 sc tiled editing: For now, revert to using the entire doc size.
626e15b sc tiled editing: Fix the conversion so that it works with zoom too.
fa9b324 sc tiled editing: Don't crash on desktop.
25d3fd0 sc tiled editing: Better setup of drawinglayer, so that images are swapped in.
3a864cd sc tiled editing: Allow turning the text selection into cell selection.
2bbce7d sc tiled editing: Never show the cell cursor.
d4e4c5e SwViewShell::PaintTile: avoid incomplete tiles when layout is not ready
b102c58 android: update emulator / debugging documentation
7e05693 sc tiled editing: Double click in the cell selects the entire cell.
69b5055 sc tiled editing: Reuse the selection method.
0f83c5d sc tiled editing: Make the 'long tap to select' work again in shapes.
1d4c727 SwEditWin tiled rendering: do word selection even if the cursor was corrected

Work Report: TDF0002 – Milestone 2

Appendix B: Calc Tiled Rendering Implementation

Calc tiled rendering was in a state that hindered implementation of the editing functionality. While technically the improvements in the tiled editing were out of scope of this work, we had to implement that anyway to be able to pass the acceptance tests:

Commit ID *Commit message*

```
d38c2cd sc tiled rendering: Split ScGridWindow::Draw() to setup and drawing.
31df8a2 sc tiled rendering: tdf#85848: Use DrawContent() in PaintTile() too.
6cf7e17 Kill this return.
a812c87 sc tiled rendering: Make the drawings and charts work.
81dd0c4 sc tiled rendering: Simplify the zoom computation.
9916032 sc tiled rendering: RefreshZoom() is called already in SetZoom().
c25062f sc tiled rendering: Don't adjust the text width according to printer.
5666552 sc tiled rendering: Avoid warnings about a non-existing SdrPaintWindow.
2384e6e sc tiled editing: EndTracking() is necessary in the tiled rendering
case.
8b5f18e sc tiled editing: Set the correct map mode for editeng.
355f119 sc tiled rendering: add missing mm100 -> twips conversion
0e3fc4a sc tiled editing: Set the viewport size to document size
4f658bf sc tiled editing: using function GetPrintArea to calculate document
size.
3aaab96 sc tiled editing: Use GetPrintArea() for FillInfo()/ScOutputData too.
0877317 sc tiled editing: Don't resize the gridwin to the entire document.
bf8b86e sc: Don't limit the zoom to some funny numbers.
9fb3ace sc tiled editing: Changed function to PixelToLogic.
```

Appendix C: 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. They implement more pleasant switching of slides, handling of hyperlinks, and add more buttons to the toolbar – like the Italics, Underline, Strike-through, and Save.

Commit ID *Commit message*

```
fd6129b add swipe support for presentation
75f5168 implement swiping gesture as a generic event
ef990c7 remove unused boolean (propagating touchEvent instead of intercepting
it)
7f5e0f8 refactor swipegesture to be handled by layerview
2ee313f tdf#89174:disable swipe gesture when zoomed in.
636e618 android: LOAbout doesn't need to extend Activity
9f2ddc5 android: cleanup LibreOfficeUIActivity, remove PreferenceEditor
26afe0a add italic/underline button with icons for different screen res
950d338 add saving action placeholder
6007813 implement strike out and .uno:Save
76d797b tdf#89705: disable Ctrl+Click for mobile apps.
bbb857c tdf#89705 hyperlink clickable in writer (not in impress, yet)
e3b517e tdf#89705: make hyperlinks clickable on impress/calc
619f033 tdf#89705: provides callback for url handling on android.
```

Appendix D: General Android and Other Bug-fixing and Cleanups

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*

```
c5d8c46 SwFmtHoriOrient: rename member variables missing their prefixes
9a93277 vcl: fix Android build
554bfc1 SdrHdlList: fix indentation in IsHdlListHit() and GetHdl()
5ea3202 SdrMarkView: aHdl -> maHdlList
e3cb10c bin/run: fix indentation
d22c2f2 sdr::overlay::OverlayManager: rmOutputDevice -> mrOutputDevice
```

libreoffice@collabora.com

☎ +44 1223 362 967



Work Report: TDF0002 – Milestone 2

bb0cae0 libreofficekit, svx, sw: fix typos
930e9fa sw: prefix global variables in edtwinn
64b3268 sw: prefix members of SwRootFrm
657b42d native-code: animcore_component_getFactory -> ctors
3097587 native-code: avmedia_component_getFactory -> ctor
35c7ad1 native-code: scd_component_getFactory -> ctors
b341722 native-code: scfilt_component_getFactory -> ctors
02b1274 native-code: sdd_component_getFactory -> ctor
87a7cec native-code: svl_component_getFactory -> ctors
2a67f6a native-code.py: Make the #if HAVE_FEATURE_SCRIPTING actually work.
1f129c8 native-code.py: VBA related services.
5c416f8 solenv: adapt native-code to partially split sd/util/sd.component
6a2f0ce clean-up ScGridWindow
21087ba GridWindow: use unique_ptr for filter box and float
e5f93bc put local functions into anonymous namespace
7ba4405 ScGridWindow: scoped_ptr -> unique_ptr
8332bc9 GridWindow: unique_ptr for overlays
a099abe GridWindow: change pNoteMarker to unique_ptr
8058189 vcl: fix Android build
f0aaeb4 HAVE_FEATURE_SCRIPTING, but no config_features.h include.
b326270 sc: no need to pass these by value

Appendix E: GtkTiledRenderer Development Test-bed

Commit ID *Commit message*

0097544 lokdocview: CursorVisible -> CursorOverlayVisible
f1e9833 lokdocview: initial LOK_CALLBACK_GRAPHIC_SELECTION handling
3851bd7 lokdocview: render graphic selection handles
e6264466 lokdocview: handle LOK_CALLBACK_HYPERLINK_CLICKED
d440dd2 lokdocview: use lok::Document::setGraphicSelection()
b2355ed71 lokdocview: implement move of images
b02216b lokdocview: return early in lcl_signalMotion()
5c3200c lokdocview: log dragging the graphic selection
e12afad lokdocview: add lok_docview_get_edit()
997ca80 lokdocview: register callback early
5fb4e35 lokdocview: add edit-changed signal, so toolbar toggle button is in sync
fdcd5a5 lokdocview.c -> lokdocview.cxx
1b33434 lokdocview: cppcheck: variableScope
60256cc lokdocview: ensure that the cursor is at least 30 twips wide
e6f7b88 lokdocview: detect click on border vs click inside graphic selection
a2675cb lokdocview: clean up quad mode
2730944 lokdocview: add lok_docview_get_document()
f7e2356 lokdocview: add LOKDocView_Impl ctor
d0e6250 lokdocview: lcl_onDestroy() -> LOKDocView_Impl::destroy()
baa0ef2 lokdocview: sync graphic move behavior with Android
a7f1ec6 lokdocview: always use zoom factor when doing twips <-> pixel conversion
2e2388d lokdocview: larger default window size
ffd4b87 lokdocview: move keyboard handling to LOKDocView_Impl
9d76382 lokdocview: move button handling to LOKDocView_Impl
986c685 lokdocview: move motion handling to LOKDocView_Impl
ffc5714 lokdocview: move cursor / selection overlay to LOKDocView_Impl
dc755a3 lokdocview: move handle rendering to LOKDocView_Impl
fa2948b lokdocview: move timeout handling to LOKDocView_Impl
aea29f6 lokdocview: move document rendering to LOKDocView_Impl
267fd94 lokdocview: GList -> std::vector
6041261 lokdocview: move CallbackData to LOKDocView_Impl
bf11827 lokdocview: move callback handling to LOKDocView_Impl
c3699a4 lokdocview: gtk_show_uri() requires Gtk 2.14
c5bd865 bin/run: Survive files with spaces in filenames.