Document Foundation Planet - Latest News

- Official TDF Blog: The Document Foundation: the name that pointed at the right thing, 16 years before (2026/06/19 06:00)
  When The Document Foundation was announced sixteen years ago, some people found the name a little flat. It didn’t sparkle. It named an object — the document — rather than a product, a movement, or an aspiration. Today, that same name is worth a second look, because it turns out to have pointed at exactly the place the digital sovereignty debate would eventually arrive. To see why, it helps to ask a simple question: when you are locked into a piece of software, where does the lock actually live? The intuitive answer is “in the application.” You feel trapped by the program — its menus, its habits, the licence you keep renewing. But the application is replaceable. You can install a different one tomorrow. What you cannot so easily replace is your documents — the years of contracts, records, reports, and correspondence you have produced. And if those documents are saved in a format that only one company’s software can fully read, then the lock was never really in the application at all. It was in the file. This is the quiet mechanism behind most document lock-in. The format does the trapping. As long as your organisation’s memory is stored in a format controlled by a single vendor, you depend on that vendor to read your own past — and that dependency does not end when you switch programs, because the documents come with you. This is also why “digital sovereignty” is not, at root, a question about geography or about which company you buy from. It is a question about control: whether you, and not a supplier, hold the keys to your own information over time. An organisation that cannot open its own archives without permission is not sovereign over them, wherever it happens to be located. The answer is older and simpler than the debate that has grown up around it: open document standards. A document saved in an open, fully published format — one any software can implement, today or in fifty years — belongs to the person who wrote it, not to the company whose program happened to create it. The format stops being a lock and becomes what it should always have been: a neutral container for your own words. The name said this all along. It put the document at the centre, because the document is where the question is decided. Sixteen years on, the rest of the conversation is catching up — and we have only just begun to scratch the surface.
- Official TDF Blog: The Getting Started Guide 26.2 has just arrived (2026/06/17 15:35)
  We are pleased to announce the release of the latest Getting Started Guide, updated for LibreOffice 26.2! The Documentation Team is proud to present this new edition, designed to help users with an introductory guide of LibreOffice, covering all aspects of the best open source free office suite, from word processing to databases as well as settings and configuration common to all modules. Writer (word processing) Calc (spreadsheets) Impress (presentations) Draw (vector graphics) Math (formula editor) Base (database management) This guide is part of our growing collection of documentation â€” lovingly written, edited, and reviewed by a global team of dedicated volunteers who are passionate about open-source software and digital freedom. The 26.2 update was coordinated by Dione Maddern, with valuable contributions from Peter Schofield and Olivier Hallot. A huge thanks to everyone involved! Dione Maddern – Guide Coordinator Ready to dive in? Download the guide for free from the LibreOffice Bookshelf Project.
- Official TDF Blog: LibreOffice releases, features, QA and accessibility – TDF Annual Report 2025 (2026/06/16 08:19)
  This is part of the Annual Report 2025 from The Document Foundation, the non-profit that coordinates the LibreOffice project and community. More will be posted soon… Releases of the Year LibreOffice’s release plan works on a time-based release schedule, with major updates every six

months (typically in February and August). So in other words, there are two new versions of LibreOffice per year. Many other FOSS projects adopt a similar time-based approach, and since 2024, LibreOffice has used a “year.month” versioning scheme – so LibreOffice 25.2, for instance, was released in the second month (February) of 2025. This versioning scheme helps users to see how old (or new) their currently installed version of LibreOffice is. In addition to the major upgrades, there were monthly smaller “point” releases, mainly fixing bugs, compatibility issues and security vulnerabilities. Major Feature Highlights LibreOffice 25.2 was released on February 6. It introduced the ability to read and write ODF version 1.4, alongside numerous interoperability improvements with proprietary OOXML documents. It became possible to automatically sign documents after defining a default certificate. Additionally, Windows 7 and 8/8.1 were designated as deprecated platforms, with support scheduled to be removed in version 25.8, and extensions and features relying on Python ceased to work on Windows 7. In LibreOffice Writer 25.2, improvements were made to Track Changes management to better handle a large number of changes in long documents. Comments were tracked in the Navigator when the focus was moved into them, while resizing the area containing comments showed a visual guide. Options were added to set a default zoom level for opening documents, which overrode the level stored within the documents themselves. It also became possible to delete all content of a specific content type, excluding headings, via the Navigator. In LibreOffice Calc 25.2, a “Handle Duplicate Records” dialog was added to select or remove duplicate records. Both the Function Wizard dialog and the Functions Sidebar deck received improvements to searching and user experience. Solver models could be saved into spreadsheets, and the Solver became able to provide a sensitivity analysis report. New sheet protection options were also added relating to Pivot Tables, Pivot Charts, and AutoFilters. Furthermore, many improvements were made to all Impress templates, which received visible elements, such as the font colour being set to black, in Master Notes and Handout. Objects could be centred on the Impress slide or Draw page in a single step, and the automatic repeating of slides could be activated in windowed mode. Finally, overflowing text in presenter notes was no longer cut off when printing. Please confirm that you want to play a YouTube video. By accepting, you will be accessing content from YouTube, a service provided by an external third party. YouTube privacy policy If you accept this notice, your choice will be saved and the page will refresh. Accept YouTube Content Meanwhile, LibreOffice 25.8 was released on August 20. It brought new performance and features to the suite. In the User Interface, the Welcome/What’s New dialog began offering access to the user interface picker and appearance options, which allowed new users to leverage LibreOffice’s flexible UI and personalise the look and feel according to their preferences. The release also provided better interoperability with Microsoft Office files, offering more accurate handling of DOCX, XLSX, and PPTX files with fewer formatting issues, thanks to changes such as: A complete overhaul of word hyphenation and spacing Font management in Impress updated to be compatible with PowerPoint files The addition of new functions in Calc: CHOOSECOLS, CHOOSEROWS, DROP, EXPAND, HSTACK, TAKE, TEXTAFTER, TEXTBEFORE, TEXTSPLIT, TOCOL, TOROW, VSTACK, WRAPCOLS, and WRAPROWS There were, of course, other important new features, such as the ability to export to the PDF 2.0 format, and several new ScriptForge library services. Please confirm that you want to play a YouTube video. By accepting, you will be accessing content from YouTube, a service provided by an external third party. YouTube privacy policy If you accept this notice, your choice will be saved and the page will refresh. Accept YouTube Content Performance Throughout 2025, the LibreOffice community continued to work on improved performance in the suite. In LibreOffice 25.2, the speed of font previews in Calc was greatly improved. Additionally, the speed of saving from XLS to ODS was greatly improved after the impact of increasing the supported number of columns to 16k, and saving ODS files with large merged ranges became faster. Spreadsheets with lots of conditional formatting opened and saved much faster, while spreadsheets with lots of comments also saved much faster. Finally, the speed of loading XLS files was greatly improved after the impact of increasing the supported number of columns to 16k. In

LibreOffice 25.8, performance was upgraded so that everything ran faster, from startup to scrolling through large documents, with significant speed improvements delivered on less powerful machines. In benchmark tests, Writer and Calc opened files up to 30% faster. Optimised memory management allowed for smoother operation on virtual desktops and thin clients. Quality Assurance For every release, the LibreOffice Quality Assurance community produced Alpha, Beta and Release Candidate versions, giving users the chance to test the software (and report bugs) well in advance of the final release. Throughout the years, thousands of bugs were confirmed, triaged and resolved. The QA team wrote monthly reports about its activity on the QA blog. Accessibility In LibreOffice 25.2, the Accessibility Sidebar featured improved warning and error levels, along with a new option to ignore specific warnings. Additionally, user interface elements were updated to report an accessible identifier that can be utilised by assistive technologies. Platform-specific enhancements included on Windows, where accessibility was automatically enabled whenever a tool querids information on the accessibility level, and accessible relations were now correctly reported. Meanwhile, on Linux, the positions of UI elements, including those on Wayland, were accurately reported on the accessibility level. LibreOffice 25.8 added an accessibility check for links and references in header/footer. Menus in the File ▸ Templates ▸ Manage Templates dialog became screen reader accessible on Windows. Support for the IAccessible2 “text-indent” attribute was added, which could be used by assistive technology like screen readers to report the indent of a paragraph’s first line. Additionally, the table design view in Base no longer became unresponsive when a screen reader was active on Windows. Finally, comboboxes and other controls inside toolbars were also represented in the accessibility tree of the application. Like what we do? Support the LibreOffice project and The Document Foundation – make a donation, or get involved and help our volunteers. Thank you!

- Official TDF Blog: Video: Join the LibreOffice community! (2026/06/15 07:14)
  LibreOffice is the free, private, open source office suite â€“ and successor to OpenOffice. It’s made by a worldwide community, and you can be part of it! Boost your skillset, learn new people, and have fun â€“ find out what you can do for LibreOffice. (This video is also available on PeerTube.) Please confirm that you want toÂ play a YouTube video. By accepting, you will be accessing content from YouTube, a service provided by an external third party. YouTube privacy policy If you accept this notice, your choice will be saved and the page will refresh. Accept YouTube Content
- LibreOffice QA Blog: QA/Dev Report: May 2026 (2026/06/12 12:04)
  General Activities LibreOffice 25.8.7 was announced on May 12 Olivier Hallot (TDF) improved the explanation for DATE function in Calc’s Function Wizard, continued documenting in help how LibreOffice treats non-integer values for function parameters expecting integer values, added a help page for Draft View in Writer, added help for Calculated
- Official TDF Blog: Euro-Office, open standards, and native ODF (2026/06/11 06:17)
  A welcome commitment to open standards — and why it should end with ODF as Euro-Office’s native document format. The Euro-Office pre-announcement has generated considerable coverage across the European press over the past few days. The Document Foundation welcomes the attention that open standards are receiving — and welcomes still more the commitment the announcement makes to them. Before the discussion settles, we would like to clarify one point and state one expectation. Several reports have described Euro-Office as “the first European open source office suite.” Reading the pre-announcement carefully, we do not find the coalition making that claim, and it is not one we would endorse. Europe has been building free and open source office software for many years: LibreOffice, developed by this Foundation and a worldwide community, is itself European, mature, and far from alone. The “first” framing appears to have emerged in the speed of a launch day

rather than in the text of the announcement. We note it not to claim precedence — precedence is not the point — but because accuracy serves the cause of open standards better than enthusiasm alone. Read on its merits, the announcement gives a great deal to welcome. The promise to improve support for the OpenDocument Format is precisely what the European free software community has long asked for, and we take it in good faith and with genuine appreciation. We have always held that sovereignty begins with the format, not with the logo on the application — and a coalition that understands this is one worth encouraging. We would also state an expectation, in the spirit of encouragement rather than demand. Improved support is a beginning, not a destination. A format that is merely supported is one a suite can read and write as a courtesy, while a native format is the one in which its documents are created, stored, and trusted across the years — and that is precisely where digital sovereignty is won or lost. The only destination consistent with the sovereignty Euro-Office invokes is ODF as its native document format. A genuinely European, genuinely sovereign office suite cannot treat the open standard as a concession to outsiders, it has to speak ODF as its mother tongue. The Document Foundation looks forward to that moment, and will be glad to acknowledge it when it comes.

- Marius Popa Adrian: Firebird Adds Unix Domain Socket Support (2026/06/06 08:20)
  Firebird is gaining support for Unix Domain Sockets (UDS) through PR #9034, a feature aimed at improving local inter-process connectivity, especially in constrained environments like Android and iOS. The change introduces a new unix:// connection string format and a RemoteServiceUnixSocket configuration option, allowing Firebird clients and services to communicate over filesystem-based sockets
- LibreOffice QA Blog: QA/Dev Report: April 2026 (2026/05/13 10:19)
  General Activities LibreOffice 26.2.3 was announced on April 30 Olivier Hallot (TDF) updated Writer’s Compatibility options help, improved documentation for wrapped images in headers, worked on documenting in help how LibreOffice treats non-integer values for function parameters expecting integer values, improved help for Declare statement in BASIC, added help for
- Marius Popa Adrian: How Firebird’s New Parallel Sort Changes Everything (2026/05/08 14:38)
  Breaking the Single-Thread Barrier: How Firebird’s New Parallel Sort Changes Everything1. Introduction: The Multi-Core ParadoxThere is a specific economic and technical frustration well-known to database architects: authorizing the purchase of high-end silicon with 64 or 128 cores, only to watch the OS scheduler show a single thread redlining while the rest of the hardware sits idle.
- Marius Popa Adrian: Help Us Test the New Firebird Docker Images (2026/04/24 10:43)
  Major Update: Help Us Test the New Firebird Docker Images We have been working on a significant overhaul of the official firebird-docker images, and a pre-release version is now available for testing at: Pre-release Container Registry We would love to get feedback from the community before these changes are merged upstream. What’s New Firebird 6
- Marius Popa Adrian: FlameRobin 0.9.16 released (2026/04/24 06:09)
  FlameRobin 0.9.16 released focuses on: modernizing CI/build tooling fixing compiler/linker issues , improving packaging (Flatpak), and delivering a set of Firebird metadata/DDL extraction and SQL editor correctness improvementshttps://github.com/mariuz/flamerobin/releases/tag/0.9.16
- Stephan Bergmann: Git file from nowhere (2026/04/22 13:17)
  Git is not only broken by design, it also has some practical shortcomings around git-format-patch and git-am, as it turns out: $ mkdir repo1$ ls -a repo1. ..$ git init -q repo1$ ls -a repo1. .. .git$ git -C repo1 commit --allow-empty -F ../subject.txt[master (root-commit) 82b1f4c] Empty test commit$ git -C repo1 log --oneline --stat82b1f4c (HEAD -> master) Empty test commit$ ls -a repo1. .. .git$ cat repo1/hello.txtcat: repo1/hello.txt: No such file or directory$ git -C repo1 format-patch -k -1 HEAD -o ..../0001-Empty-test-commit.patch$ rm -fr repo1$ mkdir repo2$ ls -a repo2. ..$

git init -q repo2$ ls -a repo2. .. .git$ cat repo2/hello.txtcat: repo2/hello.txt: No such file or directory$ git -C repo2 am -k ../0001-Empty-test-commit.patchApplying: Empty test commitapplying to an empty history$ git -C repo2 log --oneline --stat292e19c (HEAD -> master) Empty test commit hello.txt | 1 + 1 file changed, 1 insertion(+)$ ls -a repo2. .. .git hello.txt$ cat repo2/hello.txtHello from the void Which leaves the question, what's the content of that subject.txt? Want to take a guess? See below. $ cat subject.txtEmpty test commit--- hello.txt | 1 + 1 file changed, 1 insertion(+)diff --git a/hello.txt b/hello.txtnew file mode 100644index 0000000..479e903--- /dev/null+++ b/hello.txt@@ -0,0 +1 @@+Hello from the void

- Ravi Dwivedi: LibreOffice Conference Budapest 2025 (2026/04/21 03:54)
  In September 2025, I attended the LibreOffice Conference in Budapest, Hungary, on the 4th and the 5th, and a community meeting on the 3rd. Thanks to The Document Foundation (TDF) for sponsoring my travel and accommodation costs. The conference venue was Faculty of Informatics, Eötvös Loránd University (ELTE). The conference was planned to be held from the 4th to the 6th, but the program for the 6th of September had to be canceled due to the venue being unavailable because of a marathon in Budapest. So, all the talks got squeezed into just two days, making the schedule a bit hectic. The TDF had booked my room at the Corvin Hotel. It was a double bedroom with a window. The breakfast was included in the hotel booking. The hotel was walking distance from the conference venue. One could also take a tram from the hotel to reach the venue. A shot of my room. Photo by Ravi Dwivedi, released under CC-BY-SA 4.0. A tram in Budapest. Photo by Ravi Dwivedi, released under CC-BY-SA 4.0. 3rd of September On the 3rd of September, we had a community meeting at the above-mentioned venue. I walked with my friend Dione to the venue. Upon reaching there, I noticed that the university had no boundaries and gates. This reminded me of the previous year's conference venue in Luxembourg, which also had no boundaries or gates. In contrast, Indian universities and institutes typically have walls and gates serving as boundaries to separate them from the rest of the city. Many of these institutes also have security guards at the entrance, who may ask attendees to present proof of admission before allowing them inside. I was surprised to find that institutes in Europe, like the one where the conference was held, did not have such boundaries. The building where the conference was held was red, which happened to be the same color as the building for the previous year's conference venue. I remember joking with Dione that the criteria for the conference venue might have been the color of the building. The red building in the picture served as the conference venue. Photo by Ravi Dwivedi, released under CC-BY-SA 4.0. During the community meeting, we shared ideas on how to spread the word about LibreOffice. The meeting lasted for a couple of hours. After the community meeting, we went to the hotel for dinner sponsored by the TDF. These Esterházy cake bites were really yummy. Photo by Ravi Dwivedi, released under CC-BY-SA 4.0. Raspberry Currant cake slices. Photo by Ravi Dwivedi, released under CC-BY-SA 4.0. 4th of September On the first day of the conference, attendees were given swag bags containing a pad, sticky notes, a pen, a conference T-shirt, and a bottle. Conference swag. Photo by Ravi Dwivedi, released under CC-BY-SA 4.0. The talks started early in the morning with Eliane Domingos, Chairperson of TDF's Board of Directors, giving the inauguration talk. As always, I found Italo Vignoli's talk on the importance of document freedom interesting. During the snack break, I noticed that there were three types of milk available for coffee: cow's milk, lactose-free milk, and almond milk. Almond milk is rare in India, but I have managed to get it, but I have never seen lactose-free milk in India. Since I run fundraisers in my projects, such as Prav, I could relate to Lothar K. Becker's talk. He discussed the issue that certain implementations in LibreOffice require a budget that is too large for any single interested entity to fund independently. Furthermore, The Document Foundation (TDF) cannot legally receive funds from government entities. Therefore, there is no organization or entity to pool resources from all the interested entities to finance the implementation. Lothar giving his presentation. Photo by Ravi Dwivedi, released under CC-BY-SA 4.0. Another talk was by the Austrian Armed

Forces on their migration to LibreOffice. I wanted to know why they migrated, and I found out that they did it for their digital sovereignty, and not for saving on the license costs. Another point presented in the talk was that LibreOffice is available on all the operating systems, while the Microsoft Office suite is not that widely available. The migration was systematic and was performed over a few years. They started working on it in 2021, and the migration was finished recently. In addition, it also required training their staff in using LibreOffice. Presentation on migration to LibreOffice by Austrian Armed Forces. Photo by Ravi Dwivedi, released under CC-BY-SA 4.0. The lunch was inside the university canteen. We were provided lunch coupons by the TDF. I got a vegan coupon with 4000 Ft written on it, which meant I could take lunch for up to 4000 Hungarian forints. My lunch ticket for the conference. Photo by Ravi Dwivedi, released under CC-BY-SA 4.0. The lunch I had on the first day. Photo by Ravi Dwivedi, released under CC-BY-SA 4.0. During the evening, it was my turn for the presentation. I was done with preparing my slides ten days before my talk. I also got my slides reviewed by friends. My talk was finished in 20 minutes, while I was given a 30-minute slot. This helped us catch up on the schedule. Furthermore, I made my talk interactive by asking questions and making sure that the audience was not asleep. During my talk, my friend Dione took my pictures with my camera. My talk was on how free software projects could give users a say in freedom to modify the software. I illustrated this using the Prav project that I am a part of. After the talks were over, we were treated to a conference dinner at Trofea Grill. It had a great selection of desserts, which helped me sample some Hungarian desserts. The sponge cake was especially good. Desserts at Tofea Grill. Photo by Ravi Dwivedi, released under CC-BY-SA 4.0. 5th of September The next day—the 5th of September—I went with Dione to the venue early in the morning, as her talk was the first one of the day. Her talk was titled Managing Tasks with Nextcloud Deck. Later that day, I also attended a talk on Collabora. At lunch, I found the egg white salad quite tasty. Dione giving her presentation. Photo by Ravi Dwivedi, released under CC-BY-SA 4.0. Egg white salad. Photo by Ravi Dwivedi, released under CC-BY-SA 4.0. After the lunch break, we had the conference group photo. I had a Nikon camera, which we used to take the group photo. I requested a university student to take our group photo and also taught her how to operate the camera. Group photo By the evening, the conference ended, after which we went to a pub, which was again sponsored by TDF. I had beer, but that one really tasted bad, so I couldn’t finish it. The only vegetarian option was goat cheeseburger, which my friend Manish and I opted for. The burger tasted awful. Apparently, I don’t like goat cheese. The next day I went sightseeing with Dione in Budapest. Stay tuned for our adventures! Credits: Thanks to Dione and Richard for proofreading.

- Chris Sherlock: Demystifying the LibreOffice VCL Coordinate Pipeline (2026/04/19 03:05)
  I've been looking fairly closely at the coordinate system of VCL. I understood it fairly well, but as I have a Pro subscription of Gemini, I asked it to look at it for me to tell me how it thinks it works. Here is what it came up with: If you’ve ever worked on a complex UI rendering engine, you know that placing a single pixel on the screen is rarely as simple as draw(x, y). A single coordinate actually exists in several different "realities" simultaneously. In LibreOffice’s VCL (Visual Components Library), a coordinate has to travel through up to five distinct spatial dimensions to get from the mathematical purity of a document model to the physical LED pixels on your monitor. If you get an offset wrong or apply a scaling factor out of order, your text disappears off the page, your borders render fuzzy, or your PDF exports break entirely. To fix these issues and modernize the rendering stack, we have to establish a strict, predictable pipeline. Here is a deep dive into the five coordinate spaces of the LibreOffice VCL, and the math required to traverse them. The 5 Coordinate Spaces Think of these spaces as a series of nested Russian dolls. To get to the center (the document), you have to open them one by one. 1. Absolute Logic (Document Space) This is the pure, mathematical space of the document itself. Units: Defined by the MapMode (e.g., 1/100th of a millimeter for high-precision printing). Origin (0,0): The absolute top-left corner of the page or document canvas. The Variable: Represented simply as nX or nY. 2. Logic Units (Pipeline Space) This is an intermediate staging area. The

coordinate is still in logical document units, but it has been intentionally shifted. The Shift: mnOutOffLogic. Why it exists: This is an artificial shift applied to the document origin. It is frequently used when VCL needs to render a specific sub-section or "tile" of a document without actually changing the underlying coordinates of the objects themselves. 3. View Space (Viewport Space) Welcome to the realm of pixels—specifically, pixels relative to the viewport (the scrollable area of the application). The Transformation: To get here, we multiply the Logic Units by the DPI and Zoom scale (mfMapScX / mfMapScY). The Shift: mnMapOfsX / mnMapOfsY (The Mapping Offset). Why it exists: The origin (0,0) here is the top-left of your current scroll position. As you scroll down a Writer document, the mapping offset changes, shifting the view without altering the document. 4. Window Space (Client Space) These are pixels relative to the GUI window frame itself. The Shift: mnOutOffOrigX / mnOutOffOrigY (The VCL Pixel Offset). Why it exists: The origin (0,0) is the top-left corner of the specific LibreOffice window or UI widget you are interacting with. VCL uses this offset internally to account for things like scrollbars, widget borders, or docking areas inside a window. This is the coordinate space where your mouse click events natively arrive. 5. Device Space (Physical Space) The final destination. These are absolute pixels mapped to your physical hardware. The Shift: mnOutOffX / mnOutOffY (The Screen Origin). Why it exists: The origin (0,0) is the top-left corner of your physical monitor. This is the coordinate system that the underlying operating system graphics APIs (like CoreGraphics on macOS, Cairo on Linux, or GDI/DirectWrite on Windows) require to actually illuminate a pixel on your screen. The Mathematical Pipeline To safely traverse these spaces without causing "double-subtraction" bugs or off-by-one pixel errors, we chain the transitions together in a strict sequence. Here is the Forward Path (converting a Document coordinate to a Physical Screen pixel): Absolute → Logic Units: Add the logical offset.LogicUnits = nX + mnOutOffLogicX Logic Units → View: Apply the zoom scale, then add the scroll offset.View = (LogicUnits * mfMapScX) + mnMapOfsX View → Window: Add the internal VCL widget offset.Window = View + mnOutOffOrigX Window → Device: Add the physical screen position.Device = Window + mnOutOffX When handling a mouse click, we run this exact pipeline in reverse (The Inverse Path), carefully subtracting the offsets and dividing by the scale to figure out exactly which 1/100th of a millimeter the user clicked on. Why Sub-Pixel Accuracy Matters Historically, rendering engines used integer math (tools::Long) for these transitions. If a line ended up at pixel 10.7, it was truncated to 10. For basic UI elements, this was fine. However, modern graphics rely heavily on anti-aliasing (B2D rendering) and high-fidelity vector exports (PDFs). If you truncate a coordinate too early in the pipeline, you lose the fractional data. When you eventually scale that truncated coordinate back up, that tiny fractional loss multiplies into massive visual artifacts—lines appear to "shimmer" when scrolling, or text glyphs collide with each other. By upgrading this pipeline to handle high-precision double math at every stage (Sub-Pixel stages), LibreOffice can pass mathematically perfect coordinates to the OS-level drawing APIs, ensuring that your documents look perfectly crisp at any zoom level. Launch the Interactive VCL Visualizer VCL Coordinate Space Visualizer Device Space (Monitor) Coordinate Pipeline Variables UI Layout (Chrome) <label>Show Sidebar:</label><input id="check-sidebar" type="checkbox" /> <label>Show Status Bar:</label><input id="check-statusbar" type="checkbox" /> Device <-> Window (Physical) <label>Offset X:</label><input id="slider-dev-x" max="1000" min="0" type="range" value="360" /> <label>Offset Y:</label><input id="slider-dev-y" max="800" min="0" type="range" value="100" /> Total Window Bounds <label>Total Width:</label><input id="slider-win-w" max="1500" min="300" type="range" value="800" /> <label>Total Height:</label><input id="slider-win-h" max="1000" min="200" type="range" value="600" /> Window <-> View (Scroll) <div

class="control-group"><label>Offset X:</label><input id="slider-win-x" max="0" min="-1200" type="range" value="0" /> <label>Offset Y:</label><input id="slider-win-y" max="0" min="-1400" type="range" value="0" /> View <-> Logic (MapMode) <label>Scale Factor:</label><input id="slider-scale" max="3.0" min="0.5" step="0.1" type="range" value="1.5" /> Logic <-> Absolute Logic <label>Offset X:</label><input id="slider-abs-x" max="500" min="-500" type="range" value="0" /> <label>Offset Y:</label><input id="slider-abs-y" max="500" min="-500" type="range" value="0" />   VCL Internal State (CoordinateMapper) Output Bounds (mnOutWidth) : 800 px
 Output Bounds (mnOutHeight): 600 px

 VCL Target Point Pipeline 1. Absolute Logic Space: (500, 500)   + Logic<->Abs Offset: (0, 0) 2. Logic Space: (500, 500)   * View<->Logic Scale: 1.5x 3. View Space: (750, 750)   + Window<->View Offset: (0, 0) 4. Window Units: (750, 750)   + Device<->Window Offset: (0, 0) 5. Device Pixels: (0, 0)   VCL Window (Drag Me)   <canvas height="2500" id="document-canvas" width="2500"></canvas>   Properties<hr /> Character
Paragraph
Area Gallery<hr /> Arrows
Flowchart
Icons   Page 1 of 1 | 0 words | Default Style | English (Australia)

- LibreOffice QA Blog: QA/Dev Report: March 2026 (2026/04/15 14:21)
  General Activities LibreOffice 25.8.6 and LibreOffice 26.2.2 were announced on March 26 Olivier Hallot (TDF) added a help page for drag & drop features for items in text documents, updated help for Text Grid in Writer and PDF export General page and improved the help for Calc’s advanced filter options
- Ravi Dwivedi: Hungary Visa (2026/04/14 05:50)
  The annual LibreOffice conference 2025 was held in Budapest, Hungary, from the 3rd to the 6th of September 2025. Thanks to the The Document Foundation (TDF) for sponsoring me to attend the conference. As Hungary is a part of the Schengen area, I needed a Schengen visa to attend the conference. In order to apply for a Schengen visa, one needs to get an appointment at VFS Global and submit all the required documents there, which are then forwarded to the embassy. I got an appointment for a Hungary visa at VFS Global in New Delhi for the 24th of July. There were many appointment slots available for the Hungary visa. One could easily get an appointment for the next day at the Delhi center. There were some technical problems on the VFS website, though, as I was unable to upload a scanned copy of my passport while booking the appointment. I got an error saying, “Unfortunately, you have exceeded the maximum upload limit.” The problem didn’t get fixed even after contacting the VFS helpline. They asked me to try in the Firefox browser and deleting all the cache, which I already did. So I created another account with a different email address and phone number, after which I was able to upload my passport and book an appointment. Other conference attendees from India also reported facing some technical issues on the VFS Hungary website. Anyway, I went to the VFS Hungary application center as per my

appointment on the 24th of July. Going inside, I located the Hungary visa application counter. There were two applicants ahead of me. When it was my turn, the VFS staff warned me that my passport was damaged. The “damage” was on the bio-data page. All the details could be seen, but the lamination of the details page wore off a bit. They asked me to write an application to the Embassy of Hungary in New Delhi stating that I insist VFS to submit my application along with describing the “damage” on my passport. I got a bit worried about my application getting rejected due to the “damage.” But I decided to gamble my money on this one, as I didn’t have time (and energy) to apply for a new passport before this trip. Moreover, I had struck down a couple of fields in my visa application form which were not applicable to me, due to which the VFS staff asked me to fill out another visa application. After this, the application got submitted, and it was 11,000 INR (including the fee to book the appointment at VFS). Here is the list of documents I submitted: My passport Photocopy of my passport Two photographs of myself Duly filled visa application form Return flight ticket reservations Payslips for the last three months Invitation letter from the conference organizer (in Hungarian) Proof of hotel bookings during my stay in Hungary Cover letter stating my itinerary Income tax returns filed by me Bank account statement, signed and sealed by the bank Travel insurance valid for the period of the entire trip It took 2 hours for me to submit my visa application, even though there were only two applicants before me. This was by far the longest time to submit a Schengen visa application for me. Fast-forward to the 30th of July, and I received an email from the Embassy of Hungary asking me to submit an additional document - paid air ticket - for my application. I had only submitted dummy flight tickets, and they were enough for the Schengen visas I applied for until now. This was the first time a country was asking me to submit a confirmed flight ticket during the visa process. I consulted my travel agent on this, and they were fairly confident that I will get the visa if the embassy is asking me to submit confirmed flight tickets. So I asked the travel agent to book the flight tickets. These tickets were ₹78,000, and the airline was Emirates. Then, I sent the flight tickets to the embassy by email. The embassy sent the visa results on the 6th of August, which I received the next day. My visa had been approved! It took 14 days for me to get the Hungary visa after submitting the application. See you in the next one! Thanks to Badri for proofreading.

- Mike Kaganski: Our doors have never been closed to you (2026/04/13 16:41)
  Maybe I’m silly. Maybe I just can’t read what they write to me (and to other Collaborans). I read this: The Document Foundation and the LibreOffice project are open by definition and principle to all developers. Our doors have never been closed to any of you, and they never will be. … and I somehow feel that this means: “we at TDF have kicked you off of membership, but you are welcome to keep contributing, and to have a warm feeling about it after that”. Open doors? I can’t even apply for membership for more than three years from now. They have officially informed me about that – this is a link to the EML with the notice from MC; it includes my reply to their original “notification”. They write: the Membership Committee expels you from the board of trustees with immediate effect. Because you didn’t relinquished your membership immediately, we decided also considering all circumstances to block membership for at least three calendar years, thus at least up to December, 31 2029. If I had relinquished my membership as the MC asked, I would have lost my right to challenge this “temporary inconvenience” – and I am puzzled by the claim by a board member that “in the meantime … [I] can reapply for membership as soon as the legal matters have been settled.” (https://community.documentfoundation.org/t/comment-about-collabora-blog-post-tdf-community-blog/13626/9). I can re-apply, but – it is clear I will not be accepted until 2030 (the earliest possibility). After that the “bylaws” they invented this January will prevent me from e.g. nominating to BoD for two more years. Definitely honest and welcoming. (No idea how the remaining TDF members feel about the amazing fact that the board could decide and implement a restriction like that, limiting you without asking your opinion.) Well, enough of that. No more posts about TDF. It was nice, and I met many people during that period, that I hope I can continue to call friends; but the current policy of that thing

claiming nice goals and high standards is so disgusting, that I am even glad to not have relation to that anymore. Let’s do some hacking instead!

- Andreas Kainz: I’l be back (2026/04/11 09:12)
  After nearly 10 years, it’s time to start contributing to Open Source again. My Open Spurce journey begann with breeze icons for KDE, than I added breeze icons to LibreOffice. After that I made a the complete new colibre icon theme for LibreOffice which is the default for the Windows users. After Icon stuff I start with pressts, different visuals and User Interface related stuff like Notebookbar. Which bring me to Collabora Online Office were I fast switch to mobile toolbar and dark mode. After my first Open Source Journey I had a long break. Which show me, that Open Source is great. Other Community members update and improve my work. I can say, it’s awesome to see the work done within the DNA of each OSS. Now I will start again where I did my last work. Collabora Online (Desktop/Mobil/Tablet …). Why? Because I can! Thats the great benefit of OSS. Everyone can improve ist and I enjoy the Collabora Community a lot. In addition to it’s fast development, it’s that easy to make changes and contribute. Happy Hacking on any OSS you enjoy. It would be awesome to meet you at the Collabora Community.
- Mike Kaganski: The Post They Managed to Avoid (2026/04/05 12:56)
  “Ideally, we would have preferred to avoid this post.” When I read those opening words in Italo’s recent statement, “Let’s put an end to the speculation,” they stung. I don’t know if that specific post should have existed or not, but those first few words are a perfect reflection of the current TDF attitude. It is an attitude directed toward the very people who devoted large parts of their lives, their passion, and their hearts to the Foundation’s ideals. What I am missing is not that specific post that Italo wrote. What I expected—what I felt I earned—was a post that looked me in the eye. I wanted an explanation as to why I am being cast out from the Trustees after everything I’ve honestly given. I wanted to know my specific “guilt,” or why the Foundation now finds “guilt by association” to be an acceptable standard. And then—I would hope—they would publicly say: “Mike, we appreciate everything you’ve done. We deeply regret the unfortunate decisions we—not you—made over the years. But we feel this is the only path forward, and we are sorry.” But that is the post they successfully avoided writing.
- Marius Popa Adrian: PSFirebird: PowerShell Automation for Firebird on Windows and Linux (2026/04/02 08:52)
  PSFirebird is a PowerShell module focused on automating Firebird environments, databases, and common administrative workflows. The main goal is to make Firebird easier to script end-to-end without depending on a manual installer flow or a machine-specific setup. The problem is trying to solve was simple: working with Firebird in automation often means mixing shell scripts, ad hoc local installs,
- Mike Kaganski: TDF ejects its core developers (2026/04/01 10:36)
  I have nothing to add to this: https://www.collaboraonline.com/blog/tdf-ejects-its-core-developers/
- Stephan Bergmann: I like it here. Can I stay? Part 2 (2026/03/27 16:04)
  I came here due to a (decades-spanning, arguably perverse) love affair with the LibreOffice code body. Less so for a love of organizational bodies. So I mostly remained passive and watched the coup d’Ã©tat unfold at the Document Foundation. Where some folks apparently felt the need to have us all thrown out. Oh my. Should I have been more involved around the apparent issues at TDF? Maybe. But then again, I’m a naive little nerd who loves fixing dysfunctional code way more than navigating dysfunctional political setups. (And to be fair, I tried to do my duty, and did serve a term on the membership committee. Back when that was likely more pleasant than what it would be today.) Luckily, the code and the fun will most certainly live on, one way or another. Not least at https://collaboraonline.github.io/. Happy hacking, once more,sberg
- LibreOffice QA Blog: QA/Dev Report: February 2026 (2026/03/10 12:18)
  General Activities LibreOffice 26.2.0 was announced on February 4 LibreOffice 25.8.5 was announced on February 19 LibreOffice 26.2.1 was

announced on February 26 Olivier Hallot (TDF) added help for Writer’s text dragging and dropping options, Calc’s “Enter key for paste & clear clipboard” option and “Reject silently” in Calc’s Data

- LibreOffice QA Blog: QA/Dev Report: January 2026 (2026/02/11 15:15)
  General Activities Olivier Hallot (TDF) improved Writer help for hyphenation zones and controlling section visibility, fixed the help example for Calc’s SUMIF function, clarified the topic of fixed colours in the help for document themes, expanded the help for Calc’s sort options, explained in help the option for removing cross-document
- Mike Kaganski: A trick to develop a Windows-specific clipboard format support on other platforms, using UNO API (2026/02/07 16:13)
  Not too long ago, a change landed, that brought Biff12 clipboard format support in Calc v.26.2 – thanks Laurent! It was an easyhack that I authored some time ago; and Laurent volunteered to implement that long-standing missing feature. The small detail was, that the feature was Windows-specific (it is trivial to get the wanted clipboard content there, simply copying from Excel), while Laurent developed on another platform. Laurent had made the majority of work, before he was stuck, without being able to test / debug further changes. Then, he asked me, if there a way to continue on the platform he used. At that time, I answered, that no, one would need Windows (and Excel) to continue the implementation. So I jumped in, and added the rest, and in the end, we have created the change in co-authorship. But later, when part of my code turned out problematic, and I needed to fix it and create a unit test for it, I discovered a trick, that could put Biff12 data into system clipboard on any platform, without Excel – allowing then just paste, and debug everything that’s going on there. It relies on UNO API, and can be implemented e.g. in Basic: function XTransferable_getTransferData(aFlavor as com.sun.star.datatransfer.DataFlavor) as variant if (not XTransferable_isDataFlavorSupported(aFlavor)) then exit function oUcb = CreateUnoService("com.sun.star.ucb.SimpleFileAccess") oFile = oUcb.openFileRead(ConvertToURL("/path/to/biff12.clipboard.xlsb")) dim sequence() as byte oFile.readBytes(sequence, oFile.available()) ' changes value type of 'sequence' to integer XTransferable_getTransferData = CreateUnoValue("[]byte", sequence) end function function XTransferable_getTransferDataFlavors() as variant aFlavor = new com.sun.star.datatransfer.DataFlavor aFlavor.MimeType = "application/x-openoffice-biff-12;windows_formatname=""Biff12""" XTransferable_getTransferDataFlavors = array(aFlavor) end function function XTransferable_isDataFlavorSupported(aFlavor as com.sun.star.datatransfer.DataFlavor) as boolean XTransferable_isDataFlavorSupported = (aFlavor.MimeType = "application/x-openoffice-biff-12;windows_formatname=""Biff12""") end function sub setClipboardContent oClip = CreateUNOService("com.sun.star.datatransfer.clipboard.SystemClipboard") oClip.setContents(CreateUNOListener("XTransferable_", "com.sun.star.datatransfer.XTransferable"), nothing) end sub The first three functions are Basic implementation of XTransferable interface. Running setClipboardContent will prepare the system clipboard on any platform, using a trick of implementing arbitrary UNO interface using CreateUNOListener; and after that, pasting into Calc would allow to see if things work (if content of /path/to/biff12.clipboard.xlsb is pasted, as expected), and make improvements. If I knew this trick back then, I would of course share it with Laurent; but I thought I’d put it here now, so maybe it helps me or someone else in the future. (Note that application/x-openoffice-biff-12;windows_formatname="Biff12" there in the code was the name introduced by Laurent in the discussed commit; indeed, that, and the actual data in the file, would depend on the exact format that you work with.)
- Miklos Vajna: Improving deleted commented text ranges in Writer's DOCX filter (2026/02/04 07:42)
  If you have a commented text range, which gets deleted while track changes is on and you later save and load this with Writer's DOCX filter, that works now correctly. This work is primarily for Collabora Online, but the feature is available in desktop Writer as well. Motivation¶ It was already

possible to comment on text ranges. Comments were also supported inside deletes when track changes is enabled. These could be already exported to and imported from DOCX in Writer. But you could not combine these. With the increasing popularity of commenting text ranges (rather than just inserting a comment with an anchor), not being able to combine these was annoying. Results so far¶ Here is how a commented text range inside a delete from DOCX now looks like, note the semi-transparent comment hinting it's deleted: Commented text range, inside a tracked delete, in DOCX, Collabora Online As a side effect, this also fixes the behavior in desktop Writer, which crosses out deleted comments: Commented text range, inside a tracked delete, in DOCX, desktop In the past, the "is this deleted" property was not visible in the render result: Commented text range, inside a tracked delete, in DOCX, Collabora Online, old bad state And it was also bad in desktop Writer: Commented text range, inside a tracked delete, in DOCX, desktop, old bad state This required changes to both DOCX import and export: a comment could be deleted or could have an anchor which is a text range, but you couldn't have both. How is this implemented?¶ If you would like to know a bit more about how this works, continue reading... :-) As usual, the high-level problem was addressed by a series of small changes. Core side: cool#13988 DOCX import: fix missing delete flag on deleted comments with ranges cool#13988 DOCX export: fix missing delete flag on deleted comments with ranges Want to start using this?¶ You can get a development edition of Collabora Online 25.04 and try it out yourself right now: try the development edition. Collabora intends to continue supporting and contributing to LibreOffice, the code is merged so we expect all of this work will be available in TDF's next release too (26.8).

- LibreOffice Dev Blog: Validating ODF and OOXML files (2026/01/22 13:03)
  In LibreOffice development, there are many cases where you want to validate some documents against standards: either Open Document Format (ODF) or MS Office Open XML (OOXML). Here I discuss how to do that. Update: Article updated to reflect that odfvalidator 0.13.0 has just released. Open Document Format (ODF) Validation ODF is the native document file format that LibreOffice and many other open source applications use. It is basically set of XML files that are zipped together, and can describe various aspects of the document, from the content itself to the way it should be displayed. These XML files have to conform to ODF standard, which is presented in XML schemas. The latest version of ODF is 1.4, which is yet to be implemented in LibreOffice. You can find more about ODF in these links: Wikipedia – OpenDocument Open Document Format for Office Applications (OpenDocument) TC There are various tools to do the validation, but the preferred one is the ODF Toolkit Validator: ODF Toolkit website > ODF ValidatorÂ ODF Toolkit on Github Compiled binaries of ODF Toolkit can be downloaded from the above Github project: odftoolkit-0.13.0-bin.zip Then, you can use the ODF validator this way: $ java -jar odfvalidator-0.13.0-jar-with-dependencies.jar test.odt You may also use the online validator, odfvalidator.org, to do a validation. Online odfvalidator tool Please read this disclaimer before using: This service does not cover all conformance criteria of the OpenDocument Format specification. It is not applicable for formal validation proof. Problems reported by this service only indicate that a document may not conform to the specification. It must not be concluded from errors that are reported that the document does not conform to the specification without further investigation of the error report, and it must not be concluded from the absence of error reports that the OpenDocument Format document conforms to the OpenDocument Format specification. Office Open XML (OOXML) Validation MS Office Open XML (OOXML) is the native standard for Microsoft documents format. It is also a set of XML files zipped together, and conform to some XML schemas. You can find out more about OOXML here: Wikipedia – Office Open XML There are tools to do the validation, and the one is used in LibreOffice is Office-o-tron. One can use it with below command to validate an example file, test.docx: $ java -jar officeotron-0.8.8.jar ~/test.docx Office-o-tron can be downloaded from dev-www.libreoffice.org server of LibreOffice, and this is currently the latest version: Office-o-tron 0.8.8 Jar file It is worth noting that Office-o-tron can be also used to validate ODT

files. Extensions to ODF Standard To go beyond the current ODF standard, new features are sometimes introduced as “ODF extensions”, then are gradually added to the standard. You can read more in TDF Wiki: ODF > ODF Extensions > Why does LibreOffice save extended ODF by default? In these cases, you may see validation errors for such extensions. For example: test.odt/styles.xml[2,3347]: Error: unexpected attribute “loext:tab-stop-distance” test.odt/styles.xml[2,4849]: Error: unexpected attribute “loext:opacity” You may avoid such errors by using -e option, which ignores such unknown markups: -e: Check extended conformance (ODF 1.2 and 1.3 documents only) If you want to use the latest features from ODF validator, you should build ODF Toolkit from source. You can then run it with this command: $ java -jar ./validator/target/odfvalidator-0.14.0-SNAPSHOT-jar-with-dependencies.jar test.odt ODF Toolkit developers have recently (23 January 2026) published the new release 0.13. If you do not build from sources, you can use this new version which contains ODF 1.4 support. Final Words When you want to make sure that the ODT or OOXML document you generate is valid according to the standards, then you need validation. Sometimes, it is the opposite: you want to make sure that the input document is valid before processing it, or when you want to know if the problem is from LibreOffice (or other processors), or the document itself. Then, again, the validator is the right tool to use.

- LibreOffice Dev Blog: Outlook for the new year 2026 (2026/01/16 13:25)
  Happy new year 2026! I hope that this year will be great for you, and the global LibreOffice community, and the software itself! I hereby discuss the past year 2025, and the outlook for 2026 in the development blog. At The Document Foundation (TDF), our aim is to improve LibreOffice, the leading free/open source office suite that has millions of users around the world. Our work is community-driven, and the software needs your contribution to become better, and work in a way that you like. My goal here, is to help people understand LibreOffice code easier via EasyHacks and tutorials, and eventually participate in LibreOffice core development to make LibreOffice better for everyone. In 2025, I wrote 14 posts around LibreOffice development in the dev blog (4 of them are unpublished drafts). Outlook For the New Year Focus of the development blog for 2026 in this blog will be: Introducing new EasyHacks Using new C++20 constructs Difficulty Interesting EasyHacks Describing user interface creation with VCL VCL weld mechanism Various weld widgets Describing UNO Components You can provide feedback simply by leaving a comment here, or sending me an email to hossein AT libreoffice DOT org. We provide mentoring support to the individuals who want to start LibreOffice development. You are welcome to contact me if you need help to build LibreOffice and do some EasyHacks via the above email address. You may also refer to our Getting Involved Wiki page: TDF Wiki: Getting Involved in LibreOffice Development Let’s hope a better year for LibreOffice (and the world) in 2026.
- Miklos Vajna: Bullet improvements in Impress (2026/01/05 07:37)
  The bullet support in Impress got a couple of improvements recently, some of this is PPTX support and others are general UI improvements. This work is primarily for Collabora Online, but the feature is available in desktop Impress as well. Motivation¶ Probably the most simple presentations are just a couple of slides, each slide having a title shape and an outliner shape, containing some bullets, perhaps with some additional images. Images are just bitmaps, so let's focus on outliner shapes and their outliner / bullet styles. What happens if you save these to PPTX and load it back? Can you toggle between a numbering and a bullet? Can you return to an outliner style after you had direct formatting for your bullet? Results so far¶ The first case was about bullet editing of this document: Outliner shape with 3 outliner styles If you pressed enter at the end of 'First level', then pressed <tab> to promote the current paragraph to the second level, nothing happened. The reason for this was that our PPTX export was missing the list styles of shapes, except for the very first list style. And the same was missing on the import side, too. With this, not only the rendering of the bullets are OK, but also adding new paragraphs and using promoting / demoting to change levels work as expected. The

second case was about this document, where the second level had a numbering, not a bullet: Outliner shape with a numbering on the second level We only had UI to first toggle off a numbering to no numbering, then you could toggle on bullets. Now it's possible to do this change in one step. The last case was about styles. Imagine that you had a master page with an outline shape and some reasonably looking configuration for the first and second levels as outline styles: Outliner shape with two outline styles Notice how the last paragraph has a slightly inconsistent formatting, due to direct formatting. Let's fix this. Go to the end of the last bullet, which is currently not connected to an outline style, toggle bullets off and then toggle it on again. Now we clear direct formatting when we turn off the bullet, so next time you turn bullets on, it'll be again connected to the outline style's bullet configuration and the content will look better. Note how this even improves consistency: Writer was behaving the same way already, when toggling bullets off and then toggle on again resulted in getting rid of previously applied unwanted direct formatting. How is this implemented?¶ If you would like to know a bit more about how this works, continue reading... :-) As usual, the high-level problem was addressed by a series of small changes. Core side: PPTX export: fix missing non-first level list style for outline shapes sd doc model dump: allow invoking this from outside sd/ in a debugger sd doc model xml dump: show styles tdf#168559 PPTX imp: fix missing custom level list style for outline shapes tdf#168559 PPTX imp: fix missing list style for outline shapes on master pages tdf#89365 sd UI: fix transitioning from a numbered list to a bulleted list tdf#169275 sd UI: clear direct format when turning off bullet/num sd: extract FN_TRANSFORM_DOCUMENT_STRUCTURE handling to a new function sd, FuBulletAndPosition: avoid magic number for bullet toggle Related: tdf#89365 sd UI, from numbering to bullet: fix defaults Want to start using this?¶ You can get a development edition of Collabora Online 25.04 and try it out yourself right now: try the development edition. Collabora intends to continue supporting and contributing to LibreOffice, the code is merged so we expect the core of this work will be available in TDF's next release too (26.2).

- LibreOffice Dev Blog: Handling CI build failures (2025/12/04 23:10)
  After submitting a patch to LibreOffice Gerrit, one has to wait for the continuous integration (CI) to build and test the changed source code to make sure that the build is OK and the tests pass successfully. Here we discuss the situation when one or more CI builds fail, and how to handle that. Why Build and Test on CI? After you submit code to LibreOffice Gerrit, reviewers have to make sure that it builds, and the tests pass with the new source code. But, it is not possible for the reviewers to test the code on each and every platform that LibreOffice supports. Therefore, Jenkins CI does that job of building and testing LibreOffice on various platforms. This can take a while, usually 1 hour or so, but sometimes can take longer than that. If everything is OK, then your submission will get Verified +1 . CI Platforms for LibreOffice Currently, these are the platforms used in CI: Linux / GCC: gerrit_linux_gcc_release Linux / Clang: gerrit_linux_clang_dbgutil Android Viewer: gerrit_android_x86_64 and gerrit_android_arm Windows: gerrit_windows_wsl macOS: gerrit_mac Some of the tests are more extensive, for example Linux / Clang also performs additional code quality checks with clang compiler plugins. Also, UITests are not run on each and every platform. LibreOffice CI uses Jenkins Why Failures Happen and How to Fix? There can be multiple reasons for why a CI build fails, and give your submission Verified -1 . These are some of the reasons, and depending on the reason, solution can be different. 1. Your code’s syntax is wrong and compile fails In this case, you should fix your code, and then submit a new patch set. You have to wait again for a new CI build. 2. The code’s syntax is OK, but it is not properly formatted You should refer to the below TDF Wiki article and use clang-format tool to format your code properly. TDF Wiki – clang-format 3. Your code’s syntax is OK, but it logically not OK and fails some tests. In this case, you should try fixing your code logic, and run the tests that fail and make sure they pass. After that, you may send a new patch set and wait for a new CI build. 4. Your code’s syntax and logic is OK, but some machine fails for other reasons like their disk being full or other software/hardware failures or hiccups In this case, usually resuming

the build can be a good option. You may ask on #libreoffice-dev or #tdf-infra IRC rooms for such a resume, or request access, if you submit many patches. Resume build in LibreOffice CI 5. Your code’s syntax and logic is OK, but there are issues from other patches. In this case, intervention from other LibreOffice developers is needed. Informing people on #libreoffice-dev can help, and then you have to re-base your submission in case new patches fix the build issue. Final Notes The best way to know the reason of the build failure is to look into the CI log files. Sometimes it needs more detailed look to understand the issue, but sometimes the reason is easily provided on Gerrit as a comment. But, in the end your submission should have Verified +1 before it is suitable for merge in the LibreOffice code. This +1 as verified, does not guarantee that your patch will work as expected, but it is an important requirement.

- Miklos Vajna: Markdown import in Writer: the new template option (2025/12/03 08:35)
  Writer recently got a new markdown import option to take styles from a template, leading to much prettier output when converting markdown to PDF, DOCX or ODT. This work is primarily for Collabora Online, but the templating feature is available in desktop Writer as well. Motivation¶ A previous post mentioned recent improvements to the markdown import/export in Writer. But if you convert some markdown to e.g. PDF, all the headings just have the default look, wouldn't it be nice to take your organization template and add colors and other formatting there, automatically? Also, wouldn't it be nice if you could paste as markdown in COOL or copy the current selection as markdown? Which would enable all sorts of interesting use-cases, like using an external API to turn the selection into a summary or translating it to a different language. Results so far¶ Here is a sample input markdown: # heading 1 body text Here is how it looks like if you template it using the core.git sw/qa/filter/md/data/template.docx sample: PDF result: templated curl invocation for this: curl -k -F "data=@/path/to/test.md" -F "template=@/path/to/template.docx" -F "format=pdf" -o out.pdf https://localhost:9980/cool/convert-to Or example desktop command-line: soffice --infilter='Markdown:{"TemplateURL":{"type":"string","value":"./template.ott"}}' test.md While it would look like this by default: PDF result: default The other part is the PostMessage API of COOL, if you want to copy and paste as markdown. What's newly possible: Copy the current selection: set MessageId to Action_Copy and the value to {"Mimetype": "text/markdown;charset=utf-8"} Paste at the current cursor position: set MessageId to Action_Paste and the value to something like {"Mimetype": "text/markdown;charset=utf-8", "Data": "foo _bar_ baz"} You can read more about the PostMessage API in the COOL SDK. How is this implemented?¶ If you would like to know a bit more about how this works, continue reading... :-) As usual, the high-level problem was addressed by a series of small changes. Core side: tdf#169316 sw markdown import: add a TemplateURL parameter tdf#169316 sw markdown import, template: handle non-ODF formats as well cool#13468 sw markdown paste: add UNO command parameter to skip the detection Related: tdf#169251 sw markdown export: fix crash on OLE with no graphic Online side: cool#13419 convert-to template option: handle multiple streams in ConvertToPartHandler cool#13419 convert-to template option: pass it to doc broker cool#13419 convert-to template option: pass it to the kit process cool#13419 convert-to template option: more strict param name, generalize filenames cool#13419 convert-to template option: add testcase cool#13468 PostMessage API: allow copying the current text selection Want to start using this?¶ You can get a development edition of Collabora Online 25.04 and try it out yourself right now: try the development edition. Collabora intends to continue supporting and contributing to LibreOffice, the code is merged so we expect the core of this work will be available in TDF's next release too (26.2).
- Miklos Vajna: Interdependent tracked changes improvements in Writer, part 4: direct accept/reject (2025/11/04 07:11)
  Writer has some support for interdependent (or hierarchical) tracked changes: e.g. the case when you have a delete on top of an insert. See the third post for background. This work is primarily for Collabora Online, but the feature is available in desktop Writer as well. Motivation¶

Interdependent changes mean that the UI shows one type of change on top of another change, e.g. formatting on top of insert. Writer knows the priority of each type, so in case you have an insert or delete change and on top of that you have a formatting, then the UI will look "through" the formatting and work on the underlying insert or delete when you navigate with your cursor to a position with multiple changes and you press Accept on the Review tab of the notebookbar. Usually this is what you mean, but what if you want to work on the formatting at the top, directly? You can now open the Manage Changes dialog using the Manage button on the Review tab of the notebookbar and if you go to the formatting change row of the dialog, then pressing Accept there will accept the formatting change, not the insert or delete change. This is possible, because the dialog gives you a way to precisely select which tracked change you want to work with, even if a specific cursor position has multiple tracked changes. Results so far¶ Here is a sample ins-then-format.docx document from the core.git testcases, the baseline has an insertion, and part of that is covered by an additional formatting change on top: Interdependent tracked change: baseline If you just go in the middle of the document and press Accept, that will work with the more important insert change, so the result looks like this: Interdependent tracked change: default accept result But now you can also open the Manage Changes dialog, to be more specific by directly selecting the formatting change: Interdependent tracked change: direct accept via the dialog And when you accept the formatting change directly, the result will be just the insert change: Interdependent tracked change: direct accept result You can save and load the results in both DOCX and ODT, as usual. How is this implemented?¶ If you would like to know a bit more about how this works, continue reading... :-) As usual, the high-level problem was addressed by a series of small changes. Core side: tdf#166319 sw interdependent redlines: allow accept/reject for fmt on ins/del tdf#166319 sw interdependent redlines: fix redo of accept for fmt on ins/del tdf#166319 sw interdependent redlines: fix redo of direct reject for format Want to start using this?¶ You can get a development edition of Collabora Online 25.04 and try it out yourself right now: try the development edition. Collabora intends to continue supporting and contributing to LibreOffice, the code is merged so we expect all of this work will be available in TDF's next release too (26.2).

- LibreOffice Dev Blog: enumarray for better data arrays – EasyHack (2025/10/22 19:30)
  In LibreOffice C++ code, there are many cases where developers want to use string literals in their code. If these are messages in the graphical user interface (GUI), they should add them to the translatable messages. But, there are many cases where the string literals has nothing to do with other languages, and there will not be any further translations. In these cases, enumarray is helpful. Although enumarray can be used beyond string literals, for any kind of data. Using Symbolic Constants In old C code, using #define was the preferred way one could give a name to a string literal or other kinds of data. For example, consider this code: const char[] FRAME_PROPNAME_ASCII_DISPATCHRECORDERSUPPLIER = "DispatchRecorderSupplier"; const char[] FRAME_PROPNAME_ASCII_ISHIDDEN = "IsHidden"; inline constexpr OUString FRAME_PROPNAME_ASCII_LAYOUTMANAGER = "LayoutManager"; const char[] FRAME_PROPNAME_ASCII_TITLE = "Title"_ustr; const char[] FRAME_PROPNAME_ASCII_INDICATORINTERCEPTION = "IndicatorInterception"; const char[] FRAME_PROPNAME_ASCII_URL = "URL"; And also, the relevant states: #define FRAME_PROPHANDLE_DISPATCHRECORDERSUPPLIER 0 #define FRAME_PROPHANDLE_ISHIDDEN 1 #define FRAME_PROPHANDLE_LAYOUTMANAGER 2 #define FRAME_PROPHANDLE_TITLE 3 #define FRAME_PROPHANDLE_INDICATORINTERCEPTION 4 #define FRAME_PROPHANDLE_URL 5 Although this C code still works in C++, it is not the desired approach in modern C++. Using enumarrays In modern C++ code, you can use enumarry from o3tl library in LibreOffice. The above code becomes: enum class FramePropNameASCII { DispatcherRecorderSupplier, IsHidden, LayoutManager, Title, IndicatorInterception, URL, LAST=URL }; And also, the string literal definitions: constexpr o3tl::enumarray<FramePropNameASCII, OUString> FramePropName = { u"DispatchRecorderSupplier"_ustr, u"IsHidden"_ustr,

u"LayoutManager"_ustr, u"Title"_ustr, u"IndicatorInterception"_ustr, u"URL"_ustr }; Why an enumarray? The names are much more readable, as they do not have to be ALL_CAPPS, as per convention for symbolic constants in C. Their usage is also quite easy. For example, one can use [] to access the relevant string literal: - xPropSet->getPropertyValue(FRAME_PROPNAME_ASCII_LAYOUTMANAGER) >>= xLayoutManager; + xPropSet->getPropertyValue(FramePropName[FramePropNameASCII::LayoutManager]) >>= xLayoutManager; Final Notes In LibreOffice, enumarrays are not limited to string literals, and they can be used with other data. This task is tdf#169155, and if you like, you may try finding some instances in the code and modernize it using enumarrays. To learn more about LibreOffice development, you can refer to TDF Wiki. You may follow this blog to read about EasyHacks, tutorials and announcements related to LibreOffice development.

- LibreOffice Dev Blog: enum class instead of unscoped enum – EasyHack (2025/10/16 14:05)
  Since C++11 when enum class (also named scoped enum) is introduced, it is preferred to plain enum which is inherited from C programming languages. The task here is to convert the old enum instances to enum class. Rationale enum class has many benefits when compared to plain enum, as it provides better type safety among other things. Implicit conversion to integers, lack of ability to define the underlying data type and compatibility issues were some of the problems with plain enum that enum class solved in C++11. Although since then enum has improved and one can specify underlying type in the scoped enumerations. Plain enums pollute namespace, and you have to pick names that are too long, and have to carry the context inside their names. For example: INETMSG_RFC822_BEGIN inside enum _ImplINetRFC822MessageHeaderState. With an enum class, it is simply written as HeaderState::BEGIN. When placed inside a file/class/namespace that makes it relevant, it is much easier to use: it is more readable, and causes no issues for other identifiers with possible similar names., See this change: 593f08303 convert enum _ImplINetRFC822MessageHeaderState to enum class You can read more about that in: Why is enum class considered safer to use than plain enum? Finding Instances You may find some of the instances with: $ git grep -w enum *.cxx *.hxx|grep -v "enum class" When you count it with wc -l, it shows something more than 2k instances. Examples Commits You can see some of the previous conversions here, which is around 1k changes: $ git log --oneline -i -E --grep="convert enum|scoped enum" This is a good, but lengthy example of such a conversion: 9072c5c855 convert SbxFlagsBits to scoped enum Implementation First of all, please choose good names for the new enum class and values. For example, you may convert APPLICATION_WINDOW_TITLE into Application::WindowTitle. Therefore, do not use the old names as they were. Converting enum to enum class is not always straightforward. You should try to understand the code using the enum, and then try to replace it with enum class. You may need to add extra state/values for situations where 0 or -1 or some default value was used. There are cases where a numerical value is used for different conflicting purposes, and then you have to do some sort of conflict resolution to separate those cases. You may end up modifying more and more files, and a few static_casts where they are absolutely necessary because you are interpreting some integer value read from input. These are the places where you should check the values yourself in the code. You have to make sure that the numerical value is appropriate before casting it to the enum class. If you want to do bitwise operations, you should use o3tl::typed_flags, for example: enum class FileViewFlags { None = 0x00, MultiSelection = 0x02, ShowType = 0x04, ShowNone = 0x20, }; template<> struct o3tl::typed_flags : o3tl::is_typed_flags<FileViewFlags, 0x26> {} Then, you may use it like this: if (nFlags & FileViewFlags::MULTISELECTION) mxTreeView->set_selection_mode(SelectionMode::Multiple); Please note that 0x26 is the mask, and is calculated by applying OR over all possible values. All the values must be non-negative. Final Notes This is a simple development task for LibreOffice also known as EasyHack, which is filed in Bugzilla as tdf#168771. These small tasks are defined to help newcomers to LibreOffice development community to improve their skills with LibreOffice coding. You may find other instances related to C++ here: TDF Wiki: List of EasyHacks

- Miklos Vajna: Markdown import/export in Writer (2025/10/07 06:13)
  Writer recently got a Markdown import & export filter and there were a number of improvements to that. This work is primarily for Collabora Online, but the feature is available in desktop Writer as well. Motivation¶ Ujjawal Kumar contributed a markdown import to Writer, as part of Google Summer of Code (GSoC) this summer. Mike Kaganski of Collabora also created a minimal markdown export in Writer. I looked at the feature differences between the two, and filled in various gaps in the markdown export. I also added a few general markdown import/export improvements relevant for normal Writer documents, like embedded image support. Results so far¶ Here is a sample case of a document using inline code spans: Code span: baseline Exporting this to markdown & loading back to Writer, the code span was lost: Code span: old result And now it's preserved: Code span: new result This also works with code blocks. Second, here is a document with lists: Lists: baseline Exporting this to markdown & loading back to Writer, the lists were lost: Lists: old result And now they are preserved: Lists: new result This also works with nested lists. Third, here is a document with an image: Image: baseline Exporting this to markdown & loading back to Writer, the image was lost: Image: old result And now it's preserved: Image: new result This also works with embedded and anchored images. Fourth, here is a document with a table: Table: baseline Exporting this to markdown & loading back to Writer, the table was lost: Table: old result And now it's preserved: Table: new result This also works with table alignments and nested tables (to the extent the markdown markup allows that). Fifth, here is a document with a quote block: Quote: baseline Exporting this to markdown & loading back to Writer, the quote's paragraph indentation was lost: Quote: old result And now it's preserved: Quote: new result How is this implemented?¶ If you would like to know a bit more about how this works, continue reading... :-) As usual, the high-level problem was addressed by a series of small changes. Core side: desktop lok, doc save: register .md for Markdown sw markdown export: handle code tdf#168152 sw markdown export: handle lists tdf#168172 sw markdown export: handle images tdf#167564 sw markdown export: handle tables sw markdown export: handle block quote tdf#168317 sw markdown export: handle code block sw markdown export: handle table cell adjustment tdf#168341 sw markdown filter: handle links on images sw markdown export: handle line breaks tdf#168446 sw markdown export: improve image name/description/title handling tdf#167564 sw markdown export: handle multi-para table cells tdf#167564 sw markdown export: handle nested table cells tdf#168617 sw markdown filter: map tasks to checkbox content controls and back sw markdown filter: import images with 'data:' URLs sw markdown filter: export non-linked inline images tdf#168662 sw markdown export: extract inline image export functions tdf#168662 sw markdown export: handle anchored images Want to start using this?¶ You can get a development edition of Collabora Online 25.04 and try it out yourself right now: try the development edition. Collabora intends to continue supporting and contributing to LibreOffice, the code is merged so we expect all of this work will be available in TDF's next release too (26.2).
- Mike Kaganski: A fairy tale about poor UX enforcing vendor lock-in (2025/09/15 08:25)
  Once upon a time, there was a girl, who used WhatsApp in her iPhone. She was rather active there, and collected quite some important data in the app over time. But some things in her iPhone were inconvenient; and the phone was slowly aging. So she wanted to change her phone some day. For her birthday, a fairy, who learned somehow about the girl’s wish, presented her a new Android phone. That was a nice new phone, and the girl was so happy! She decided to move everything from the old phone to the new one immediately. She was worrying about how to move the precious data between the devices; but she felt a huge relief, when the phone spoke: “The fairy told me how important your data is to you; and I have magic powers to handle it all. Just connect the old phone to me with a cord”. So she did. The new phone started its work; and the girl could see how the progress bar was gradually moving to completion; but suddenly it stopped; minutes passed, but the bar was motionless. The girl was impatient to start using her new shiny device, but she knew that she needs to wait. And she waited; and waited; but after an hour

passed, she noticed something horrible: the old phone was sucking the life out of the new device through the cable! The scared girl could only hope that the process would resume, and finish before the new phone is out of power. She searched and learned, that iPhones are known for their insatiable hunger, and whenever they are connected to anything with energy, they start sucking it. She couldn't even ask the new phone to shine less brightly to save the energy – because it wasn't ready for such things yet. She used her wireless charger, but its powers were fewer than the hunger of iPhone, combined with the hard work done by Android. The energy level still decreased too fast. In the end, when the hope almost vanished, the progress resumed moving! But immediately, the new phone said: "When I collected your data from your old phone, something bad happened, and I failed to collect something. I will continue, but please check later, what's missing!". Only a couple of energy drops were remaining in the new phone, when it finished its task, and could be disconnected from the vampire. But the girl was terrified, when she opened WhatsApp, connected to it (using a magic SMS confirmation), only to see that all her data is lost! She tried to open WhatsApp on the old phone to check if something is still there, and saw that the app had disconnected her. So she used the SMS magic again, and – to her great relief – everything was there! She askes WhatsApp, how to move the data; and it answered, that if she moved from iPhone to iPhone, or from Android to Android, she could use a backup; but from iPhone to Android, only the Transfer Wizard was supported. So she decided to try again. Long story short, but this time, everything repeated exactly the same. The energy was sucked from the new phone; the wireless charger couldn't fully compensate that; the progress stopped, and then a failure happened; the data wasn't there. This time, when she spelled the SMS magic, she needed to wait some minutes before it worked. It was because the wise powers out there were caring and guarded her from possible villains trying to steal her data, so demanded a delay. The girl was desperate. She was almost ready to throw the new phone away. But after some time, she decided to talk to WhatsApp again. She asked it, what to do, and got the same advice. She explained her problem, but the app was adamant. And only after a long persuasion, and even some threats, the app told her a secret, that there are third-party paid apps, that can also move her data from phone to phone! Poor girl had no choice, and bought one such app. She launched it, and asked to transfer her data. And the helper app said: "Connect your phone to your old iPhone with a cable!" You can imagine how sad was the girl hearing that. But she did what the app asked; and as she feared, the iPhone started to do what it always did. The progress was painfully slow, as you already guessed. Everything was almost exactly as before. But something changed this time: there was no error! The task took even longer; and when it finished, the new phone almost died; but it finished!!! The heart of the girl was full of happiness. She wanted to open WhatsApp immediately, to know if everything is there! But first, she had to do the SMS magic. She casted the spell … and the powers replied her, that she has to wait eight hours! I lack the ability to describe her anger, when she heard that. She came through pains, she lost her money, lots of time and nerves – and now she couldn't do the last step just now. The time lasted incredibly slow … but eventually, she overcame that last obstacle, and was glad to learn, that this time, everything was there. But I hear the demonic laughter of someone, who designed a process, where one insanity was piled upon another: where you can't move the data using normal means; where you use a vampire cables; where error messages don't allow you to fix anything by telling where the problem is; where you have to pay to have your data back (oh no, WhatsApp is not like that ransomware, just the end result is the same); where the security measures aggravate the grief, because they don't account for problems of their own software; and overall, where the app makes its transfer so complicated, that people would rather stay with old vendor, just to not experience that again.

- allotropia: Collabora and allotropia merge (2025/05/28 10:20)
  This deal unites the largest team of corporate Office engineers to deliver on Collabora Productivity's mission to restore Digital Sovereignty to its users, while making Open Source Office Rock. It supercharges Collabora's Online Office products and services portfolio with rich German

language capability, deeper experience of vertical applications, new Web Assembly skills, and a wider unified partner ecosystem. Through improved product richness this sharpens the competitive edge of FLOSS Office productivity against mass-market proprietary alternatives. CAMBRIDGE, UK – May 28th 12:00 CEST – 2025 Collabora Productivity, the world's leading provider of collaborative Open Source Office editors have completed a merger with allotropia. Collabora has invested heavily in building Collabora Online (COOL) – a market leading, on-premise, secure, interoperable, open-source solution for document editing and collaboration deployed to any modern browser. This is complemented by desktop and mobile apps across Linux, Windows, Mac, Android, iOS and Chrome-OS. Collabora provides support subscriptions to enterprise customers worldwide via a network of hundreds of trusted partners. This is now augmented by allotropia's partner and customer base. Together with our partners we deliver document and productivity excellence integrated with our partners product and service offerings. allotropia's expertise around Web Assembly combined with Collabora Online will we expect, in time, enable customer use-cases such as well as office-as-component embedding scenarios in vertical applications as well as off-line and end-to-end encrypted editing, and. This work builds on some visionary prototype funding from the Bundesministerium des Inneren (BMI) for a collaboration between the companies to enable the use of Collabora Online off-line in the browser. Further details of product investment, and direction will be announced and decided in workshops with our key customers and partners at our annual COOL Days conference in Budapest next week where staff, community and our customer and partner-ecosystem meet, swap ideas, and hear about the latest work in our upcoming major release featuring improved performance, usability, interoperability and much more. "Collabora is excited to welcome each member of the allotropia team today!" said Michael Meeks, CEO, Collabora Productivity, "We are excited to work together to accelerate our product development, enjoy our first COOL Days together, and plan the next features and possibilities to delight our customers." Collabora has invested in building a network of hundreds of partners and is approaching one hundred million docker image downloads of its document editing server software, with millions of paying users of its products, all of whom will start to benefit from this merger from today.We expect to bring the experience that allotropia has from it's relationship with CIB around vertical desktop applications (Fachverfahren) to help partners and customers migrate their Windows & Microsoft Office based business process to easy to deploy multi-platform web applications. "With our awesome team of engineers, and our WebAssembly know how, we can add significantly to Collabora's powerhouse of Office engineering prowess & their product offerings", says Thorsten Behrens, CEO of allotropia, "we've worked with them as partners for many years, and align perfectly in our goals to make Open Source office rock!" allotropia's skills in supporting and contributing to the LibreOffice code-base in Germany strengthens and unifies popular shared partner products such CIB Office and Nextcloud Office. A larger team will accelerate development and improvement of Collabora Office based products, while providing an even deeper pool of support resources to rapidly respond to customers' needs. Together we want to pay tribute to the vast legacy of those who have worked so hard to preserve and improve the source code that we depend on from Sun Microsystems, Oracle, SUSE, RedHat, IBM, TDF, Canonical, and many more, as well as the innumerable volunteer community contributors who make the Collabora Online and LibreOffice ecosystem so rich and interesting: thank you allowing us the privilege of working alongside you as we revolutionize the office productivity world together.All of our code is open source and available to the public on GitHub. Join the Collabora Online Community, take part in easy hacks and discussions in the forum. Please also see our new parent company's mirror announcement!

- Ravi Dwivedi: Libreoffice Conference 2024 in Luxembourg (2025/03/14 16:18)
  Last year, I attended the annual LibreOffice Conference in Luxembourg with the help of a generous travel grant by The Document Foundation (TDF). It was a three-day event from the 10th to the 12th of October 2024, with an additional day for community meetup on the 9th. Luxembourg

is a small country in Western Europe. It is insanely wealthy with high living standards. After going through an arduous visa process, I got to the country on the 8th of October. Upon arriving in Luxembourg, I took a bus to the city center, where my hotel — Park Inn — was located. I deboarded the bus at the Luxembourg Central station. Before walking towards my hotel, I stopped to click a few pictures of the beautiful station. All the public transport in Luxembourg was free of cost. The experience of being in Luxembourg was as if I had stepped in another world. The roads had separate tracks for cycling and separate lanes for buses, along with wide footpaths. In addition, the streets were pretty neat and clean. Luxembourg's Findel Airport. Photo by Ravi Dwivedi. Released under the CC-BY-SA 4.0. Separate cycling tracks in Luxembourg. Photo by Ravi Dwivedi. Released under the CC-BY-SA 4.0. A random road in Luxembourg with separate lane for buses. Photo by Ravi Dwivedi. Released under the CC-BY-SA 4.0. The conference venue was in Belval, while I stayed in the city center. Even though my stay was 20 km from the conference venue, the commute was convenient thanks to free of cost train connections. The train rides were comfortable, smooth, and scenic, covering the distance in half an hour. Moreover, I never found the trains to be very crowded, which enabled me to always get a seat. This is what trains look like in Luxembourg. Photo by Ravi Dwivedi. Released under the CC-BY-SA 4.0. The train ride from my hotel to the conference venue had some scenic views like this one on the way. Photo by Ravi Dwivedi. Released under the CC-BY-SA 4.0. A tram in Luxembourg with Luxembourg Central station in the background. Photo by Ravi Dwivedi. Released under the CC-BY-SA 4.0. My breakfast was included in the hotel booking. The breakfast had many options. It had coffee and fruit juices, along with diverse food options. Some of the items I remember were croissant, pain au chocolat, brie (a type of cheese), scrambled eggs, boiled eggs, and various types of meat dishes. Other than this, there were fruits such as pears. That circular pie in the center of the image is brie - a type of cheese - which I found delicious. Photo by Ravi Dwivedi. Released under the CC-BY-SA 4.0. Pre-conference, a day was reserved for the community meetup on the 9th of October. On that day, the community members introduced themselves and their contributions to the LibreOffice project. It acted as a brainstorming session. All the attendees got a lovely conference bag, which contained a T-Shirt, a pen and a few stickers. I also met my long time collaborators Mike, Sophie and Italo from the TDF, whom I had interacted only remotely till then. Likewise, I also met TDF’s sysadmin Guilhem, who I interacted before regarding setting up my LibreOffice mirror. Lovely swag bag. Photo by Ravi Dwivedi. Released under the CC-BY-SA 4.0. The conference started on the 10th. There were 5 attendees from India, including me, while most of the attendees were from Europe. The talks were in English. One of the talks that stood out for me was about Luxchat — a chat service run by the Luxembourg government based on the Matrix protocol for the citizens of Luxembourg. I also liked Italo’s talk on why document formats must be freedom-respecting. On the first night, the conference took us to a nice dinner in a restaurant. It offered one more way to socialize with other attendees and explore food at the same time. A slide from Italo's talk on document freedom. Photo by Ravi Dwivedi. Released under the CC-BY-SA 4.0. Picture of the hall in which talks were held. Photo by Ravi Dwivedi. Released under the CC-BY-SA 4.0. On the 11th of October, I went for a walk in the morning with Biswadeep for some sightseeing around our hotel area. As a consequence, I missed the group photo of the conference, which I wanted to be in. Anyway, we enjoyed roaming around the picturesque Luxembourg city. We also sampled a tram ride to return to our hotel. We encountered such scenic views during our walk. Photo by Ravi Dwivedi. Released under the CC-BY-SA 4.0. Another view of Luxembourg city area. Photo by Ravi Dwivedi. Released under the CC-BY-SA 4.0. The conference ended on the 12th with a couple of talks. This conference gave me an opportunity to meet the global LibreOffice community, connect and share ideas. It also gave me a peek into the country of Luxembourg and its people, where I had good experience. English was widely known, and I had no issues getting by. Thanks to all the organizers and sponsors of the conference!

- allotropia: ZetaJS: Combining Writer & Calc (2025/03/06 10:30)

Weâ€™ve added a great new Vue.js-3 ZetaJS demo (source)! It showcases word processing and spreadsheets inside a single web app. Calc is being used as a data source for an HTML app, filling letter templates in Writer. You can even upload custom data spreadsheets or document templates! And have you seen the nice Writer toolbar, all done with Vue.js? Weâ€™ve also updated the existing demos, showcasing Chrome PWA support with the Ping Monitor demo – just click the little install button at the top-right of the address bar, to get the Ping Monitor â€œinstalledâ€ on your desktop! Talks Meanwhile, our team was giving some great talks about our work for ZetaOffice and LibreOffice. Why not check out the recordings during your lunch break? ZetaJS & ZetaOffice Moritz Duge – OSXP Paris: LibreOffice as Web Component – moving customized office workflows into the browser Moritz Duge – 38C3 CCC congress: LibreOffice WASM & JS – Blending a C++ FOSS into a web app Stephan Bergmann: LOWA, In Need Of a VCL Plug (description) Thorsten Behrens – Distributed real-time collaboration for Writer – a first prototype (description) FOSDEM LibreOffice DevRoom talks Balazs Varga – Introducing Glow Effect for texts in shapes (description) Gabor Kelemen – Testing the QA instructions (description) Sarper Akdemir – LibreOfficeâ€™s Python API: Working around limitations of the Pythonic approach (description) Gabor Kelemen – Exploring the deprecated parts of LibreOffice API (description) News clippings Look, we made some headlines! TheRegister was following up some earlier coverage about the WebAssembly port, after Thorsten gave Liam a demo during FOSDEM. Read up the full article here. Next up In case youâ€™re around, meet us in two weeks at the FOSSAsia Summit in Bangkok, where Sarper Akdemir will give an update over our work. Dates are March 13-15. If youâ€™re based in Europe, you might instead enjoy Thorstenâ€™s talk at the Chemnitz Linux Days (Germany) from March 22-23. Looking forward to meet you there! Feedback appreciated! Please subscribe to our Newsletter or on Mastodon and let us know how you liked ZetaJS and the demos! If youâ€™re playing with the code leave a star at the ZetaJS repo or if you hit any issues please file a report on GitHub. Or just leave a comment and let us know directly – thanks for reading!

- LibreOffice Design Blog: New Templates For You – Your Feedback Matters! (2025/03/03 13:33)
  By Ndidi Folasade Ogboi For the past two months, Iâ€™ve been working on adding more templates to LibreOffice Writer as part of my Outreachy project. My goal has been to create functional templates that users need the most. I created these templates based on what you told us in our survey and your response was incredible!…