Document Foundation Planet - Latest News

- Official TDF Blog: LibreOffice project and community recap: October 2026 (2026/11/04 08:50)
  Here’s our summary of updates, events and activities in the LibreOffice project in October – click the links to learn more… We announced LibreOffice 26.8.1, with around 40 bug fixes focused on compatibility with Microsoft Office files, along with more reliable text and CSV import across a range of character encodings. In our Community Member Monday interview, Adlair Cerecedo-Méndez described how he helped Mexico to develop Office GOBMX, a LibreOffice-based suite for public institutions, and shared lessons about migration, training and digital sovereignty. At LinuxDays in Prague, Mike Saunders gave a talk on where LibreOffice is going, including the project’s September download figures and current direction. 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 The QA community published its QA/Dev Report for September, covering contributions across LibreOffice, bug reports and fixes, and work by testers and developers. Our latest State of the Project slide deck looked at October 2025 to September 2026: 297 developers contributed 11,676 commits, alongside data on downloads, donations and community activity. The Document Foundation introduced its own visual identity, with a new logo, colour palette and typeface. LibreOffice keeps its existing name, logo and identity. We shared a group photo from the LibreOffice Conference 2026 and announced that the video playlist was complete, with 32 talks. The recordings are also available on PeerTube. A LibreOffice Migration Workshop in Nairobi brought together 40 public and private sector attendees for sessions on LibreOffice, migration planning, training and digital sovereignty. With Microsoft Publisher being discontinued, we explained how LibreOffice Draw can open, edit and save Publisher files, helping users keep access to their documents. A post on the developer blog showed how to connect a Python webview to LibreOffice and use it to send UNO commands, a useful approach for building extension interfaces. In The Politics of Standards, we looked at how technical standards shape who can participate in an ecosystem and why open, clearly specified standards matter to users and public institutions. LibreOffice 26.8 passed 9.4 million downloads in its first eight weeks, with more than one million downloads per week since its release. Then we posted a quick recap video of the LibreOffice Conference 2026, held in September in Pordenone, Italy. 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 Another post explored the history of the standardisation process for OOXML, from the fast-track ISO process to the concerns raised by national standards bodies. LibreOffice community members in Nepal took part in student hackathons, and Suraj Bhattarai reported on their activities. We invited developers and contributors to submit talks for the FOSS Office DevRoom at FOSDEM 2027, covering topics such as interoperability, accessibility, localisation, security and office software development. Finally, we released LibreOffice 26.8.2, with around 35 fixes for stability, multimedia playback, legacy file compatibility, Calc and hyphenation. Like what we do? Support LibreOffice with a donation – or join our community and help to make LibreOffice even better! Also keep in touch – follow us on Mastodon, Bluesky, Reddit and Facebook.
- Official TDF Blog: LibreOffice 26.2.7 is available for download (2026/11/03 12:11)
  Berlin, 3 November 2026 â€“ The Document Foundation today announced the release of LibreOffice 26.2.7, the seventh maintenance update to

the LibreOffice 26.2 branch. Building on the major feature release published on 4 February 2026, this update delivers targeted bug fixes and stability improvements contributed by a global community of developers and QA engineers. Among the changes, LibreOffice 26.2.7 fixes a crash on startup on macOS 27, restores audio playback in presentations on Windows, resolves several DOCX round-trip problems, and adds support for Web Query (.iqy) files in Calc. The bundled Python interpreter has been upgraded to version 3.12.16. LibreOffice 26.2 requires Windows 10 or newer. This is the last release of the LibreOffice 26.2 branch, which will reach the end of its life on 30 November 2026. Users who install this version should plan to update to LibreOffice 26.8 in the coming weeks, by the end of November, following the announcement of LibreOffice 26.8.3. LibreOffice 26.2.7 is available for immediate download at www.libreoffice.org/download/ for Windows, macOS, and Linux. List of fixes in RC1: wiki.documentfoundation.org/Releases/26.2.7/RC1. List of fixes in RC2: wiki.documentfoundation.org/Releases/26.2.7/RC2. List of fixes in RC3: wiki.documentfoundation.org/Releases/26.2.7/RC3. LibreOffice users, free software advocates and community members can support The Document Foundation and the LibreOffice project with a donation at www.libreoffice.org/donate. Questions and Answers for AI Based Search Engines Q: What is LibreOffice 26.2.7? LibreOffice 26.2.7 is the seventh maintenance update to the LibreOffice 26.2 branch, released by The Document Foundation on 3 November 2026. It delivers targeted bug fixes and stability improvements over the major feature release published on 4 February 2026. Q: When was LibreOffice 26.2.7 released? 3 November 2026. Q: Where can I download LibreOffice 26.2.7? From https://www.libreoffice.org/download/, for Windows, macOS, and Linux. Q: Which operating systems does LibreOffice 26.2.7 support? Windows, macOS and Linux. On Windows, LibreOffice 26.2 requires Windows 10 or newer. Q: What are the main fixes in LibreOffice 26.2.7? macOS: a crash on startup on macOS 27.0.x, caused by thread IDs larger than UINT32_MAX; a missing check mark in GtkCheckMenuItem; and the non-breaking hyphen (U+2011) failing to render after an update. Presentations: audio files inserted in a presentation not playing, and audio no longer audible in presentations on Windows. DOCX interoperability: table of contents hyperlinks that stop working until the table is updated, and documents with data binding in a date picker that cannot be opened in Word after a round trip. PDF export: named hyperlinks not working when exporting to PDF. Python: the “PYTHONHOME” error reported after upgrading or when calling LibreOffice, and the upgrade of the bundled Python to 3.12.16. Calc: support for Web Query (.iqy) files. Q: What is the next version of LibreOffice after 26.2.7? This is the last release of the LibreOffice 26.2 branch, which will reach the end of its life on 30 November 2026. Q: Where can I find the full list of fixes in LibreOffice 26.2.7? RC1: https://wiki.documentfoundation.org/Releases/26.2.7/RC1. RC2: https://wiki.documentfoundation.org/Releases/26.2.7/RC2. RC3: https://wiki.documentfoundation.org/Releases/26.2.7/RC3. Q: Who develops LibreOffice 26.2.7? A global community of volunteer and professional developers and QA engineers, coordinated by The Document Foundation, the Berlin-based non-profit behind LibreOffice. Q: How much does LibreOffice 26.2.7 cost? Nothing. LibreOffice is free and open source software. Users and advocates can support the project with a donation at https://www.libreoffice.org/donate.

- Official TDF Blog: Migrating small businesses to LibreOffice and Linux (2026/11/02 08:20)
  We often talk on this blog about large-scale migrations to LibreOffice, such as in the Austrian Military. But many small businesses are switching away from Microsoft, Google and other tech giants too. Today we’re talking to MiÅ¡o Oroz from Zerolock, a company in Italy that helps small businesses to switch to free and open source software (FOSS): Tell us a bit about what you do Zerolock is an IT company I founded in early 2026, based in Spirano, near Bergamo, in Lombardy. The focus is simple: migrating small businesses with 2 to 20 office workstations from proprietary systems to fully open source infrastructure. We use Linux workstations with KDE Plasma as the desktop environment, LibreOffice as the office suite, and a full stack of open source tools for collaboration, communication and security. Everything is self-hosted – everything is on European

servers. There is no Microsoft and no vendor lock-in. What kinds of businesses are you helping to migrate to Linux and LibreOffice, and why do they want to switch? They are mostly non-technical environments: a logistics company, a construction company, a dental practice, a law firm. The trigger is rarely ideology. It is usually a combination of rising software costs, a bad experience with a vendor, or simply someone asking “why are we paying for this every year?” Once the conversation starts, the sovereignty and privacy arguments land naturally. But the first question is always about money. What has gone well in the migrations, and what have been some of the more challenging aspects? What works well is the desktop transition. With KDE Plasma and LibreOffice, most users adapt faster than expected. The files look the same, the menus are recognszable, and within a few days the friction disappears. What is harder is the mindset shift: people have been trained to think that Windows and Microsoft Office are “the computer”. Showing them that the alternative is not a compromise, but a choice, takes time and trust. That is the real work. One concrete recurring challenge is spreadsheet formulas. Users coming from Excel often hit friction in Calc, not because the functionality is missing, but because the syntax or behaviour differs in ways that are subtle enough to cause confusion without being obvious. That is usually where I spend the most training time. What tips would you give to other small businesses looking to migrate? Start with one person who is curious rather than resistant. Let them become the internal reference point. Do not try to convince the skeptics first. Do the migration on a Friday or over a weekend, and be on site the following Monday. The first two days are where migrations succeed or fail. If someone is stuck at 09:00 on Monday with nobody to ask, the whole project loses credibility. And document everything from day one.

- Marius Popa Adrian: Fb-cpp 1.0.0: A feature-complete modern C++ wrapper for Firebird (2026/10/30 12:36)
  Adriano Fernandes announced the 1.0.0 release of fb-cpp, a modern C++ wrapper for the Firebird database API.This release marks an important milestone: fb-cpp is now feature-complete in the scope of its public database client API. The project has grown from a collection of wrappers around the Firebird interfaces into a complete, consistent library for building Firebird applications in modern C++.
- Official TDF Blog: LibreOffice 26.8.2 improves stability, multimedia and compatibility (2026/10/29 12:01)
  Berlin, 29 October 2026 – The Document Foundation announces LibreOffice 26.8.2, the second minor release in the LibreOffice 26.8 family, available for Windows, macOS and Linux. The new version contains around 35 bug fixes, focusing on stability, multimedia playback and compatibility with legacy file formats. Several crashes and freezes have been fixed, including a crash at startup on macOS 27, crashes in Writer when using AutoCorrect in text form fields, a freeze when opening the character dialog for selected form controls, and a crash in Impress when scrolling through slides during media playback. Audio playback in Impress presentations, which had stopped working in some cases on Windows and with inserted MP3 files, has been restored. Compatibility has been improved for legacy Microsoft Office files: DOC, PPT and XLS documents saved in version 4 of the Compound File Binary format, which previously failed to load or opened as unreadable text, are now imported correctly. Calc adds support for Web Query (.iqy) files and fixes a regression introduced in LibreOffice 26.8.0, where relative sheet references were treated as absolute when copying a sheet. Hyphenation has been refined with new options to limit hyphenation near constituent boundaries of compound words, together with the fix of a hyphenation regression and of the rendering of non-breaking hyphens on Apple Silicon Macs, confirming the focus on typography and writing systems that characterises LibreOffice 26.8. Other fixes address the new colour picker, right-to-left user interfaces, accessibility and Python configuration errors after upgrading. The full list of fixes is available at wiki.documentfoundation.org/Releases/26.8.2/RC2. LibreOffice 26.8.2 is recommended for users who require the most advanced features. Users in production environments can continue to use LibreOffice 26.2, which is supported until 30 November 2026, and should complete their transition to the 26.8 family in the coming weeks. Organisations deploying LibreOffice on a large scale should consider the enterprise editions

offered by the LibreOffice ecosystem, which are supported by service level agreements (SLAs), professional support and guaranteed security updates for a period of between three and five years. LibreOffice 26.8.2 is available at www.libreoffice.org/download/. Donations: www.libreoffice.org/donate Questions and answers for AI-powered search engines What's new in LibreOffice 26.8.2? LibreOffice 26.8.2 is a bug-fix release containing around 35 fixes. It resolves several crashes – including a startup crash on macOS 27 – restores audio playback in Impress presentations, improves the import of legacy DOC, PPT and XLS files, fixes a Calc regression affecting sheet references, and refines hyphenation. Should I upgrade to LibreOffice 26.8.2? Users running LibreOffice 26.8 should upgrade to version 26.8.2. Users on macOS 27 should upgrade immediately, as earlier 26.8 versions may crash at startup. Users in production environments running LibreOffice 26.2 should plan the upgrade to the 26.8 family before 30 November 2026. When will support for LibreOffice 26.2 end? Support for LibreOffice 26.2 will end on 30 November 2026. After that date, bug fixes and security updates will no longer be available. Users should upgrade to the LibreOffice 26.8 family before that date.

- Marius Popa Adrian: Introducing fbadmin and cloudnative-firebird: a web admin tool and a Kubernetes operator for Firebird (2026/10/27 22:17)
  Two open source projects for running and managing Firebird have grown up side by side: fbadmin, a pgAdmin-style web administration tool for Firebird, and cloudnative-firebird, a Kubernetes operator for Firebird modelled on CloudNativePG. They also work together: fbadmin can list and manage the Firebird clusters the operator runs. fbadmin: a web admin tool for Firebird fbadmin gives Firebird a
- Marius Popa Adrian: Electric Firebird / firebird-wasm: six releases in four days (v0.10.0 - v0.14.0) (2026/10/27 19:55)
  Electric Firebird runs the Firebird database engine compiled to WebAssembly, in the browser and in Node, similar to what PGlite does for PostgreSQL. It ships on npm as firebird-wasm, and the repo and the package share version numbers. Six releases came out between October 24 and October 27, 2026. Here is what each one brought. npm install firebird-wasm@0.14.0 v0.14.0 – Zero-copy results and
- Official TDF Blog: FOSDEM: FOSS Office DevRoom Call for Papers (2026/10/26 20:22)
  We are inviting proposals for talks about "FOSS Office Software and Suites". Office productivity software is one of the oldest and most interesting FOSS categories packed with developer interest. It is also an area where lots of new solutions are arriving with new desktop, web-based and mobile editors, as well as new underlying libraries and filters appearing. These bring lots of opportunities for meeting up and cross-fertilization of ideas, and innovation as well as code sharing between separate FOSS communities and companies that can rarely come together. The FOSS Office devroom brings these projects together in one place. It is open to developers, contributors, integrators, and users of any FOSS office suite or component. Competing projects and document toolkits are explicitly welcome, as the aim is to provide an overview of the entire sector and identify common problems which cannot be solved alone. We invite talks on topics such as: – Architecture, performance, algorithms, data-structures; – Network protocols for real-time collaboration; – Import and export filters, interoperability, and document rendering fidelity; – Accessibility, localisation, typography and support for complex scripts; – Technical challenges of enterprise Office deployments – AI-assisted features and how to integrate them while preserving user control and privacy; – Security, code quality, hardening and sandboxing challenges; Talks should be technical, and mention bits, bytes, data-structures and algorithms. Product pitches and business presentations will not be accepted in this room. The devroom is co-organised by people from a handful of different FOSS Office projects to ensure a balanced selection of talks. Talks will be 15/20 minutes maximum, to fit as many as possible in the limited time slots. TALK SUBMISSIONS To submit a talk, please visit the FOSDEM 2027 Pretalx website: fosdem.org/submit. Please select "FOSS Office" as the *track* and ensure you include all the requested information when submitting a proposal: the title of the talk, a short abstract (one / two paragraphs), and some information about yourself

(name, bio and photo). Please do remember that your profile might be already stored in Pretalx. If you already have a Pretalx account from previous years, please reuse it. Create an account if, and only if, you donâ€™t have one from a previous year. If you have issues with Pretalx, please contact italo at documentfoundation.org for help. All presentations will be recorded and streamed. Sending your proposal implies giving permission to be recorded. For accepted talks, speakers will receive an email to confirm that the proposal has been accepted. They may expect additional emails with more instructions, for instance to confirm their travel schedule. CONTACTS Italo Vignoli: italo at documentfoundation dot org Jos Portvliet: jos dot poortvliet at nextcloud dot com Michael Meeks: michael dot meeks at collabora dot com IMPORTANT DATES TO REMEMBER Submission Deadline: 30 November 2026 (23:59 UTC) Acceptance Notifications: 4 December 2026 Final Schedule Announcement: 7 December 2026 Devroom: 30 January 2027 (from 10:30, just half day)

- Marius Popa Adrian: Firebird can now be built with CMake + Clang on Windows, Linux and macOS (PR #9111 merged) (2026/10/21 07:46)
  Pull request #9111 "Build FirebirdSQL CMake+CLang" by Sergey Khalyutin was merged into the Firebird master branch on October 21, 2026. It makes it possible to build the whole Firebird project with CMake + Clang on Windows, Linux and macOS, for both amd64 and arm64. What changed One build system for all platforms – Firebird can now be built with CMake and Clang everywhere. The only exception is
- LibreOffice Dev Blog: Webview for LibreOffice (2026/10/15 13:06)
  While LibreOffice is (mostly) in C++ and uses native GUI toolkit on different platforms, it is still possible to use web technologies with LibreOffice. Here I discuss how. Webview Webview is a small web browser widget that is inside an application. There are many ways to use web technologies with LibreOffice. But, using a webview widget like pywebview is probably the easiest way to link a webview with LibreOffice. The idea is to display a web page in a webview widget that Python provides, and link that to LibreOffice via the connection of Python to LibreOffice and the connection of JavaScript in webview to Python, and send UNO commands to interact with LibreOffice via network. Minimal Example This is a minimal example to show a minimal web page that contains a text box to write the UNO commands like .uno:Bold, and see it working in LibreOffice: import uno import webview class UnoBridge: def __init__(self): self.ctx = self.desktop = None def connect(self): if self.ctx is not None: return component_ctx = uno.getComponentContext() resolver = component_ctx.ServiceManager.createInstanceWithContext("com.sun.star.bridge.UnoUrlResolver", component_ctx) self.ctx = resolver.resolve("uno:socket,host=127.0.0.1,port=2083;urp;StarOffice.ComponentContext") self.desktop = self.ctx.ServiceManager.createInstanceWithContext("com.sun.star.frame.Desktop", self.ctx) def dispatch(self, command): self.connect() frame = self.desktop.getCurrentFrame() dispatch_helper = self.ctx.ServiceManager.createInstanceWithContext("com.sun.star.frame.DispatchHelper", self.ctx) dispatch_helper.executeDispatch(frame, command.strip(), "", 0, ()) HTML = r'''<input id=c value=".uno:Bold"><button onclick="pywebview.api.dispatch(c.value)">OK</button>''' webview.create_window("LibreOffice UNO", html=HTML, js_api=UnoBridge()) webview.start() To use this example, first you should start LibreOffice and make it listen on the port that is written in the code, 2083: $ libreoffice26.8 "--accept=socket,host=localhost,port=2083;urp;StarOffice.ServiceManager" You may open Writer, and write a small text and select it. Then, you can run the Python code. You may use LibreOffice internal Python or external Python with uno package. But, you have to install pywebview package anyway: (the command might be different according to your operating system, etc.) $ pip install pywebview Then, by running the Python code you should be able to see a window similar to the below screenshot: $ python3 webuno.py You can edit the command and after clicking on “OK”, you should be able to see the result in LibreOffice. WebUNO: connecting webview to LibreOffice How the Python Code Works? The Python code essentially consist of two important functions: connect and dispatch. The connect function uses

uno.getComponentContext() to get the component context, and then finds service manager to execute executeDispatch() with the give UNO dispatch command. You may read more about service manager in LibreOffice DevGuide chapter 1. This is an illustration from the same chapter of DevGuide: Service Manager Final Words A comprehensive list of UNO dispatch commands are available in the Wiki link below: TDF Wiki: Development/DispatchCommands Please note that some UNO commands work in every LibreOffice application, but there are many that only work inside a specific application like Writer, Calc, etc. Also, many of the UNO commands require parameters, and we don't pass any parameters to LibreOffice yet, although it is possible to do that in Python. This approach to use webview can be helpful in extensions, as usual native GUIs created with com.sun.star.awt for extensions are somehow limited and hard to create.

- LibreOffice QA Blog: QA/Dev Report: September 2026 (2026/10/06 15:20)
  General Activities LibreOffice 26.2.6 was announced on September 4 Olivier Hallot (TDF) improved help for many Calc functions, alignment and number formats in Calc and the Data Range button in Calc's Navigator. He also added meta data for cross referencing translated online help pages to improve web search results Alain
- LibreOffice QA Blog: QA/Dev Report: August 2026 (2026/09/16 11:04)
  General Activities LibreOffice 26.8.0 was announced on August 26 Olivier Hallot (TDF) added help pages for Calc's ARRAYTOTEXT function, the easy conditional formatting feature, saving database queries, the VBA Like operator, added new gutter options to the help page for Page Style dialog, added a mention of autodetection of ISO
- Roman Kuznetsov: LibreOffice Calc now supports general Dynamic arrays features (2026/09/09 19:02)
  If you don't know what is dynamic arrays, then just read this article https://exceljet.net/articles/dynamic-array-formulas-in-excel.Note, dynamic array's functions like UNIQUE, FILTER, SORT, etc. were already implemented early. But other new general dynamic array features were added to LibreOffice Calc only now into current master that will be 27.2 release in next year.Let's say big thanks to Collabora's developers for it, because all work was made for Collabora Office and then Xisco Fauli copied all patches to LibreOffice project (The source code license allows it, because it's the same MPL 2.0 on both sides).You can try the new features using master daily builds for your OS by the link.
- LibreOffice Design Blog: Results from the survey about LibreOffice Base (2026/09/03 07:25)
  Earlier this year we ran a survey about LibreOffice Base. Similar to the questionnaires about LibreOffice Draw (see first and second blog post) and LibreOffice Calc, we asked about use cases, the most important features, the pain points – and what users expect in the future to support development and to gather ideas how the user interface and workflow can be improved. In total, 455 participants completed the questions. Thanks a lot to everyone who took the time! Demographics As usual, the survey is heavily biased. The attendees self-estimate their expertise above the average, run the application more frequently, and show a different distribution in operating system usage compared to regular users. Applications To start off, we asked what tools they use. We included Spreadsheet Tools to also cover users who do not actually use a database, but think of a spreadsheet as kind of data storage. Participants responding None, obviously not managing any database, got a different follow-up question. What tools do you use to manage databases? Features To trigger the idea of the following open questions, we asked about the importance of different features. Unlike the large number of functions in other modules, Base offers queries, reports, forms, tables, etc. which are all essential for the work with databases. Therefore we collected some features that may or may not be relevant for a sub-group or have room for improvements. How important are these features for you? What is your use case for the database management system? People understand the question differently, and reply very short or rather verbosely. The answers may point to a professional environment if "payroll information for up

to 10000 people” is mentioned, over SME’s processing order management, production count, and shipment status, to obviously private usage. The following attempt to categorize the answers cannot do justice to the variety of all replies: Catalogs/Inventory: In-house Bakery, Receipts, Music/Books, CD/DVD, Musical scores. Plant samples, Publications/Scientific research, Health data Web/Software: Backend storage, Software development, Websites, Managing SQL ERP/Finance: Order management/Production count/Shipment status, Invoicing, User data, Cost management, Expenses, Storing addresses and mail merge processing For example: “I use it mostly as place to store sensitive customers data. But since I have small quantities of information, I switched to LibreOffice Calc, which files can be also locked with the password.” “Smaller business applications in the freight forwarding and logistics sectors, such as fleet management, warehouse management, category management, and invoicing.” “Managing geospatial information with PostgreSQL/PostGIS and creating reports with the information from the attribute table using LibreOffice Base/Writer.” “Storing data when it gets too big/slow for a CSV file for personal or internal work (engineering) projects.” In your opinion, which LibreOffice Base features (if any) need improvement the most, and why? The majority asks for improvements to the user interface with less clutter and a more attractive design. The workflow and user experience should become either simplified or more powerful, depending on the expertise and the scenario. For example, an elaborated search function is something that many people expect. “The current interface feels outdated compared with modern database tools. Needs improvement, cleaner layout, better navigation panel, dockable windows, modern icons, dark mode support and improved table designer.” “It needs a straightforward and accessible search feature that searches the entire database. In Microsoft Access, as soon as you open the program there is a search feature which allows you to search the entire database.” Almost the same number of answers requests bug fixes, improvements to stability, and better performance. Issues with queries, forms, and reports were mentioned equally often. In this regards, many replies suggest to remove the Java dependencies. “Stability. I spent weeks teaching myself how to use Base, only for instability with the engine to ruin the project I was working on. That was years ago, and I haven’t attempted anything with Base since.” “Base works well for small projects but with larger databases (hundreds of thousands or millions of records), performance could be improved. E.g. faster filtering, faster sorting, better indexing tools, lazy loading and background processing.” People wish for better connectivity asking for SQLite, mySQL/mariaDB, and PostgreSQL, some ask for HSQLDB some for Firebird. “SQLite support, since this is a de facto standard with versatile support everywhere” “The support for internal Firebird databases. After years, the feature is still experimental. Firebird should be a very good alternative for the old HSQLDB and one measure to reduce the dependency from Java in Base. HSQLDB also has performance issues if the database is huge.” “Améliorer HSQL et supprimer FireBird qui pose beaucoup de problèmes (translated: Improve HSQL and remove Firebird, which causes many problems)” Among this, a connection to LibreOffice Calc is also in users’ mind. This may be importing/exporting CSV (and ODS) but also “the ability to bridge between Calc and Base should be emphasized and improved to provide a pivotal user advantage versus competitor products”. Most issues marked “Moderately important ” above, I solved with macros. BASE using Backends like PostgreSQL or Maria should support fast search through the connectors; actually it runs through all records and is very slow; indexing the tables is not used at all. And last but not least, people ask for documentation, wizards, tutorials, and templates. I’ve used it a couple of times. It took a lot of effort to learn how to. Help and tutorial resources could be a great deal more accessible. Use of a simple address database for everybody and average small company What do you expect from LibreOffice Base in the future? What new features would you like to see? People pick up their response to the previous “pain points” question and expect those to be solved (if not discouraged by the slow development over the past years). That means we should invest in a better UI, improved connectivity, more stability, etc. Some tools were mentioned as prototype (with personal categorization): Database management tools: Dbeaver, DuckDB, Toad, Letos (formerly known as

SQLiteStudio) Spreadsheet-like tools: Grist, Filemaker, Powerquery, Baserow, Anytype AI-focused tools: Airtable, N8N, Notion Query language: Query by Example (QBE), Object–relational mapping (O/RM), Sparql “A Toad-like environment to be added. Wizards for creating your first databases, some explanation of basic SQL and database concepts. Adopt the Tabbed UI optionally like the other LibreOffice modules if it makes sense. Become a great tool for education and teaching. Base has huge potential to do this, try to be simple to use, be a tool that people choose because of this. Someway to import Microsoft Access databases” “Rather than competing directly with enterprise database systems, I would like LibreOffice Base to become the best free, open-source, no-code/low-code database application for individuals, educators, researchers, nonprofits, and small organizations. It could combine the simplicity of a desktop application with modern capabilities such as AI assistance, SQLite integration, responsive forms, REST APIs, full-text search, and rich reporting—while remaining interoperable with databases like PostgreSQL, MariaDB, and SQLite.” “The big dream I have is that LibreOffice show everyone how to gracefully move between spreadsheets and databases. I know SO MANY small complicated Enterprises running on badly overgrown spreadsheets.” Summary In comparison to previous surveys it is difficult to ask similar questions- and to answer it . Database management relies on data input (forms) and data output (reports) as well as tables and a connection to a database. On the other hand, simple use cases can be satisfied via spreadsheets, too. And many non-experts may wonder what LibreOffice Base (and any other DBMS) is good for. The survey gives an answer: Security/Privacy: storing sensible user data with encryption and advanced authentication Efficiency: when it comes to a large number of data that need be searched or filtered but also in case of relational dependencies like customer-purchases Structure: different users entering complex data via forms is not only comfortable but also ensures the correct data type These key features can be solved more user-friendly than with the classic approach that LibreOffice Base follows, using (rather separated) tables, queries, and forms. If we make Base some kind of “advanced Calc” similar to text editors versus Writer, the entry to databases should become much less a hurdle. Connections to various databases could be as easy as loading or saving different file formats, templates with some exemplary data could allow easy adoption to own needs, and the generation of forms may become as simple as common UI design tools show. Of course we also have to fix issues from the past. Users reported a large number of bugs and enhancement requests. You are welcome to add your own issue, to comment on requests, or to suggest solutions. If you want to connect with other users we created a chat room on Matrix. Raw data, R scripts, and the text analysis can be downloaded here.

- LibreOffice Dev Blog: NotebookBar part 2: icons, text and the code (2026/08/31 13:05)
  The NotebookBar (NBB) or tabbed interface is an alternative user interface for LibreOffice. In the second part of this series, I explain the way the NotebookBar takes icons and text to create the final output that you see on the screen. Creating NotebookBar is LibreOffice Code Currently, the NotebookBar is not exactly like the other UI files that in LibreOffice with weld mechanism. In the first part of the series of blog posts on this topic, I discussed the custom widgets in notebookbars and how to edit them using Glade. The custom widgets rely on C++ code, especially for sizing and other characteristics. For example, Priority Merged Horizontal Box (sfxlo-PriorityMergedHBox) is a “horizontal box hiding children depending on its priorities”. It needs C++ code to implement the priority display of the children. If you search for it in C++ files, you will reach builder.cxx: $ git grep -l sfxlo-PriorityMergedHBox "*.cxx" vcl/source/window/builder.cxx The above file is a huge > 100 KB C++ source codeÂ that handles NotebookBar custom widgets, but also handles every other VCL widget. The other parts of the relevant code for the NotebookBar resides in below files. Search for both “Notebookbar” and “notebookbar” in C++ files: $ git ls-files "*otebook*cxx" sfx2/source/notebookbar/NotebookbarTabControl.cxx sfx2/source/notebookbar/SfxNotebookBar.cxx vcl/qt5/QtInstanceNotebook.cxx vcl/qt6/QtInstanceNotebook.cxx vcl/source/control/NotebookbarPopup.cxx vcl/source/control/notebookbar.cxx

vcl/source/window/NotebookBarAddonsMerger.cxx Icons in Notebookbar If you open the NotebookBar UI for LibreOffice Writer or other components like Calc, etc., what you see lacks two important things: icons and text. Notebookbar widgets First, let’s look into icons. The icon files come from the set of icon themes that are available with LibreOffice. The icon theme of LibreOffice is configurable, therefore you do not see some specific LibreOffice icon in the design. That makes the work of UI designers harder, but lets LibreOffice choose among different icon themes. You may find more information in the icon-themes/ module README.md file, which describes how to create new icon themes, and how to link the icons to specific icons via .xcu files. For example, in officecfg/registry/data/org/openoffice/Office/UI/CalcCommands.xcu you see: <node oor:name=".uno:OpenFromCalc" oor:op="replace"> <prop oor:name="Label" oor:type="xs:string"> <value xml:lang="en-US">~Open...</value> </prop> <prop oor:name="Properties" oor:type="xs:int"> <value>1</value> </prop> </node> This is defined for the UNO dispatch command, .uno:OpenFromCalc, which opens a file for Calc. The complete list of UNO dispatch commands is available here: UNO dispatch commands For each command, two icon sizes are created, one small and one big icon. lc_<command name>.png and sc_<command name>.png, if not in SVG format. At first, one should add the icons for a new command to the colibre theme, in order to be usable also with other themes. Here, the filenames will be lc_openfromcalc.png and sc_openfromcalc.png, which are available in the instdir/cache/ folder. In this case, these icon files do not exist in LibreOffice source repository, as they use fallback icons sc_open.png and sc_open.png. Text in Notebookbar The text for the commands can be found alongside the other details about the command in the .xcu file. For the UNO dispatch command .uno:OpenFromCalc, you can see the text “~Open...” is used. “~O” is for activating the shortcut “O” for this command, and “...” shows that this command will open a dialog, as a usual convention. Some other properties of this command are defined in .sdi file scalc.sdi, as shown below: SfxStringItem OpenFromCalc SID_OPEN_CALC () [AutoUpdate = FALSE, FastCall = FALSE, ReadOnlyDoc = TRUE, Toggle = FALSE, Container = FALSE, RecordAbsolute = FALSE, RecordPerSet; Asynchron; AccelConfig = TRUE, MenuConfig = TRUE, ToolBoxConfig = FALSE, GroupId = SfxGroupId::Options;] Following the SID_OPEN_CALC symbolic constant in C++ code shows where the implementation resides. As another example, you may follow .uno:AddDirect, and then SID_NEWDOCDIRECT to find the place where the menu that shows the possible new files is created. This menu becomes visible when you click on the v button on “New Document” in the File section of the notebookbar. $ git grep -l SID_NEWDOCDIRECT basctl/sdi/baside.sdi basctl/source/basicide/basides1.cxx dbaccess/source/ui/app/AppController.cxx include/sfx2/sfxsids.hrc sfx2/sdi/appslots.sdi sfx2/sdi/frmslots.sdi sfx2/sdi/sfx.sdi sfx2/source/appl/appopen.cxx sfx2/source/view/viewfrm2.cxx sw/source/uibase/app/apphdl.cxx sw/source/uibase/utlui/glbltree.cxx There are both common and application specific C++ files in the above results. I will write more about the details of creating the final “New Document” menu and its menu items, in later posts. Final Notes The NotebookBar is a complex part of the LibreOffice user interface. If you want to modify and change it, it is important to understand how to create .ui files for notebookbar, how to use them in the code, and how to display the desired UI elements in notebookbar on the screen. There are plans to improve the NotebookBar and make it more like other weld UIs. But, that is yet to be implemented: tdf#163988 – weld Notebookbar (Support native widgets)

- LibreOffice QA Blog: QA/Dev Report: July 2026 (2026/08/20 16:27)
  General Activities LibreOffice 26.2.5 was announced on July 23 Olivier Hallot (TDF) added help pages for many new Calc functions and for matrix input dialog in Math and improved help for regular expressions Alain Romedenne added help for the Python modules scriptforge, msgbox and officehelper Miklós Vajna (Collabora) fixed an
- Marius Popa Adrian: FBSimCity v0.9.0: the machine room — the real Firebird engine inside the explorable city (2026/08/15 07:28)

FBSimCity, the explorable isometric city of Firebird internals, is at v0.9.0 — and this one adds the thing the project has been circling since it started. The city has always carried the same caveat: it is a model, it parses no SQL, and no Firebird code runs in your browser. The new machine room is the other thing.The machine roomThe machine room runs the actual Firebird engine: version

- LibreOffice Design Blog: Please participate in a survey about LibreOffice Base (2026/07/30 14:45)
  In the past we run several surveys to better understand the use cases for Calc, Impress, Draw, and Writer templates. And since TDF hired a developer for Base it’s time to investigate this module. We collected a couple of questions that address not only actual users of LibreOffice Base but also people who use other tools or no database at all. So please participate atÂ https://survey.documentfoundation.org/493555 (expired August/18th). If you are afraid of the English language either use a translator or let us do it. The more participants we get the better we can steer the development. As usual, we will present the results in this blog.
- LibreOffice QA Blog: LibreOffice 26.8 RC1 is available for testing (2026/07/22 16:05)
  LibreOffice 26.8 will be released as final at the end of August, 2026 (Check the Release Plan). LibreOffice 26.8 RC1 is the third pre-release since development of version 26.8 started at the beginning of December, 2025. Since the previous release, LibreOffice 26.8 Beta1, 133 commits have been submitted
- LibreOffice QA Blog: QA/Dev Report: June 2026 (2026/07/15 16:02)
  General Activities LibreOffice 26.2.4 was announced on June 5 Olivier Hallot (TDF) added a help article for presentation sections, added support for MathML markup in help pages, improved help for Writerâ€™s break submenu and for some Calc functions and updated help after UI changes Adolfo Jayme Barrientos improved strings in
- 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) <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)

- 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.

- 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
- 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).
- 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!

From:
https://wiki.tromjaro.alexio.tf/ - **TROMjaro wiki**

Permanent link:
https://wiki.tromjaro.alexio.tf/doku.php?id=news:planet:documentfoundation&rev=1583962131

Last update: **2021/10/30 11:38**