

KDE Planet - Latest News

- [Week 4 Update: KeepSecret Import/Export Feature Completed](#) (2026/06/21 17:12)

This week, the import/export feature for KeepSecret was completed and merged in !33. The implementation went through several changes before reaching its final form. My initial approach was to use a KWalletManager-compatible XML format so that data could be exchanged directly between the two applications. To match KWalletManager's structure exactly, I referenced kwalleteditor.cpp from the KWalletManager repository and implemented support for elements such as `<secret>`, `<secretType>`, `<secretContent>`, and `<secretAttributes>`. The final format now preserves all item information: -Secrets are stored as Base64-encoded data and decoded directly from the raw byte array, avoiding UTF-8 conversions that could corrupt binary or non-UTF-8 secrets. -The `contentType` field is preserved, matching the Secret Service specification's combination of raw secret bytes and content type metadata. -Every attribute key and value is exported exactly as stored, without assuming any particular schema. -Exported files use a dedicated `.keepsecret` extension instead of a generic XML extension. My first version queried libsecret directly during export. While functional, this wasn't consistent with how `CollectionModel` already manages data. Based on feedback, the final version loads secrets and attributes into the in-memory item list (`m_items`) when a wallet is opened, and the export process simply reads from that existing data. This keeps the implementation cleaner and aligned with the rest of the model. There was also a small but important UI change. Initially, the Import and Export actions were placed in the application's global drawer. During review, Marco Martin noted that these operations belong to a specific wallet rather than the application as a whole. Following that feedback, the actions were moved to the wallet page itself and integrated into the `ActionCollection` system, alongside actions such as Lock and Delete Wallet. As a bonus, they automatically gain keyboard shortcut support through the existing shortcut infrastructure. The next step is a follow-up merge request that adds support for importing the legacy KWalletManager XML format. This will make migration from KWallet to KeepSecret much smoother for existing users. The merge request!34 is open and under review.

- [Vi Mo\(tiva\)tions](#) (2026/06/21 16:31)

It's been quite a while since I last wrote an entry in this blog. It's also been some time in which my invent gitlab graph has been feeling a bit emptier and I've also been more absent from the KDE work and community in general. Several causes have been lining up to cause this lack of time and energy (or, in modern terms, bandwidth) I've had lately to dedicate to this amazing community and projects. But one could also summarize this to a lack of motivation. In order to improve the situation, I've had to resort to a last desperate measure: LEARNING VI!. I know most people reading this might want to discourage me from such a painful path, more even at my 40+ age. But I've found myself again piked by the spark of curiosity, and wanting to learn and try new things. Even writing this paragraphs feel a bit easier as I'm punching characters in INSERT MODE. Of course, I've not jumped into a vim/neovim terminal editor, but the very same Kate editor I'm used and love to work with. Just activated the integrated VI-mode and started getting comfortable with it. On this process I've also discovered several small issues that were not fully working, not implemented, or differ from the regular VIM experience. And as a KDE developer/enthusiast what's more enticing than pull up my sleeves and try to fix those. As we usually advice newcomers: better stratch one's own itch. And what's best, I'm enjoying the experience of simultaneously learning some new VI magic, seeing how it is implemented in Kate, and using it to potentially fix bugs or implement new

functionality. I'm aware that the VI-mode is not supposed to be a full replacement or mimick of the traditional terminal VIM experience, and it shouldn't be. But I think there is still enough room for improvement within that limit. For now, my first junior jobs on the matter have been implementing counted undos/redos (ex. 3u to undo the last 3 actions), and save some space in the always crowded status bar. Also trying to fix how 3rd-level symbols on my keyboard (those accessed with AltGr) are processed on Windows. But since I've discovered VI registers, I'm having the most fun fixing some usecases, adding new tests and implementing some of the missing ones. And I have also some new exciting (but more involved) ideas in mind. Still not sure if many users and members of the community will benefit from these. Are you a happy VIM (or VI adjacent) user? Did you know Kate had this mode completely integrated with its UI at a keystroke's distance? (try Ctrl-Alt-V)? Moreover, since the VI-mode code is implemented in the KTextEditor framework, any application that uses it, such as KWrite or KDevelop (or even your own app) can use this mode too! This is fun!

- [New post](#) (2026/06/20 16:32)

Hello everyone, i am uploading this blog nearly after a month. I got the wifi security QML working and one of my MR got merged in production. Here is my detailed work done during that phase Refactored the Security 8021x. I have detailed explanation in my MR description. MR - https://invent.kde.org/plasma/plasma-nm/-/merge_requests/566 Similarly, I refactored the cpp code for the wifisecurity part (WPA2, WPA3 Personal networks), MR - https://invent.kde.org/plasma/plasma-nm/-/merge_requests/582 New QML designs and QML classes (which are still in review) for Wifi Security Tab and its combo box. MR - https://invent.kde.org/plasma/plasma-nm/-/merge_requests/583 I have refactored almost 50% of the main KCM editor (the left side pane is completed) MR - https://invent.kde.org/plasma/plasma-nm/-/merge_requests/584 What's Next? I'm currently working on the remaining Network Connection tabs, such as General, Status, and others. There's still a lot of work ahead, but the progress so far has been encouraging. I'll continue sharing updates as development progresses.

- [Week 4: Channel Colors, Reset, Point Values, and Ghost Curves](#) (2026/06/20 09:57)

This is a weekly update from my Google Summer of Code 2026 project with KDE, improving effect widgets in Kdenlive, a free and open source video editor. This week JB reviewed the per-channel tabs work on MR !887 and asked for four usability improvements. All four are now in the MR. Per-channel curve colorsWith four tabs (All/R/G/B) it wasn't obvious at a glance which channel you were editing. JB suggested passing a color through the XML: `<parameter type="av_curve" name="av.g" default="" color="#44CC44"> <name>G</name> </parameter>` The color attribute is now read in AssetParameterModel and passed down to KisCurveWidget, which uses it to draw that channel's curve line instead of the default neutral color. All/R/G/B use #CCCCCC/#FF4444/#44CC44/#4488FF. Reset button for the active channelA small button next to the point spinboxes resets only the currently active tab's curve back to identity, going through KisCurveWidget::reset(); the same model update path as a normal edit, so it persists correctly on save. Input/Output value displayJB asked for the selected point's X/Y to be shown numerically, like GIMP's curve editor, for finer adjustments and reproducibility. The underlying DragValue spinboxes for this already existed in the curve widget's .ui file but had never actually rendered, for either frei0r.curves or avfilter.curves; a layout ordering bug going back years. Rather than untangle that legacy code, I added two new QDoubleSpinBox widgets directly, wired to the existing currentPoint signal that fires on point selection/drag. Ghost overlay of edited channelsJB also suggested copying GIMP's behavior of showing all non-null curves at once, so you can see how channels relate while editing one. When a channel's curve is active for editing, the other channels are now drawn faintly, in their own colors, if they've been edited away from identity. The active channel stays fully opaque on top. All four changes are in MR !887. 48 unit test assertions pass; also fixed a test-environment issue where avcurvetest wasn't loading the Kdenlive XML effect definitions, which had been masking proper av_curve

parameter typing. JB's m_widgets null-placeholder feedback from last week is still unresolved, waiting on his direction before touching it.

- [New Craft cache 26.05 published](#) (2026/06/20 09:00)

A new Craft cache has been published about a week ago. The update has already been rolled out to KDE's CD and Windows CI with the update to Qt 6.11 being the most important change. Changes (highlights) General We added updated our base image to build the Linux cache from AlmaLinux 8.10 to AlmaLinux 9.8. This implies a newer minimum required GLIBC. We did initial work on MSVC 2026 support, but given that it is not supported by Qt 6.11, we did not complete it yet. Blueprints OpenSSL 3.6.2 Qt 6.11.1 New Qt modules QtGraphs and QtQuick3D FFmpeg 8.1.1 FFmpeg is build with libass and x264 on MSVC now tiff 4.7.1 libffi 3.5.2 About KDE Craft KDE Craft is an open source meta-build system and package manager. It manages dependencies and builds libraries and applications from source on Windows, macOS, Linux, FreeBSD and Android. Learn more on <https://community.kde.org/Craft> or join the Matrix room [#kde-craft:kde.org](#)

- [KDE Goals - Call For Submissions](#) (2026/06/20 03:20)

In 2015, some members of the KDE community began a conversation about where KDE was heading and how it would get there. That conversation became the seed of what is now KDE Goals - a community initiative to help guide KDE's efforts around shared objectives. KDE Goals reaches its fifth cycle at a special milestone: KDE turns thirty this year. Thirty years of building software that puts people in control of their digital lives and enjoy freedom and privacy. The fourth cycle brought three goals that built toward this vision. Streamlined Application Development Experience focused on making KDE development smoother and more joyful. We Care About Your Input aimed to deliver a seamless out-of-the-box experience for input devices used by artists, gamers, and people with accessibility needs. KDE Needs You! worked on building a sustainable community growth targeting individuals and institutions alike. Checkout the one-year recap at Akademy last year. Now, as that cycle draws to a close and launches us into the next upward spiral, it is time once again to ask: what matters most to KDE right now? What should the community rally behind for the next two years? The floor is open. Dream big. The fifth cycle begins! Join in Anyone passionate about KDE and has a vision to share is welcome to submit a proposal. You do not have to be a developer to participate. Read the selection process carefully before you send your proposal. Got any doubts? Join our Matrix room or leave a comment at the KDE forum. Timeline Call for submissions - June 20 to August 8 Refinement of proposals - August 9 to August 27 Voting period - August 28 to September 11 Tallying and preparation - September 12 to September 18 Announcement at Akademy - September 19

- [This Week in Plasma: 6.7 is Here!](#) (2026/06/20 00:00)

Welcome to a new issue of This Week in Plasma! This week we released Plasma 6.7, and the reviews are overwhelmingly positive so far! But the team didn't sit on its laurels; instead we spent the week fixing a few issues that snuck past our beta testers or that developers didn't manage to fix in time for the final release. So far it's looking like this was a pretty smooth release. And in addition, features and UI improvements started to merge for Plasma 6.8! Check it all out: Notable new features Plasma 6.8 Monitors are now identified with color-coded number badges in all the places they're configured, making it easier to identify which one is which especially when using two monitors from the same product line. (Ramil Nurmanov, kscreen MR #456 and kwin MR #9090) Frameworks 6.28 KRunner-powered search fields can now convert between watt-hours, kilowatt-hours, and other similar energy units. (Levin Lhoest, kunitconversion MR #88) Notable UI improvements Plasma 6.7.1 There's now a smooth blend effect when switching between global themes, just like there already was when switching between color schemes. (Kai Uwe Broulik, plasma-workspace MR #6199) Plasma 6.8 Switched to the KDE-style color picker throughout QML-based KDE apps, instead of the one provided by Qt which doesn't respect KDE color schemes. A few apps had already hardcoded the KDE one, but now it happens universally so all

benefit from it. (Tobias Fella, KDE Bugzilla #520987) When moving a window by dragging it while the Meta key is held down, it is now brought in front of all other windows. (Vlad Zahorodnii, KDE Bugzilla #520339) Trying to delete a file on the desktop that you don't have permission to delete now at least shows you an error message explaining what the problem is, instead of doing... nothing. (Tobias Fella, KDE Bugzilla #515869) Made the "Connect" button in the VPN connection dialog a lot more visually obvious. (Kai Uwe Broulik, plasma-nm MR #600) Added more symbols to the "press and hold a key to see related characters" feature of the Plasma virtual keyboard. (Jonas Harer, plasma-keyboard MR #133) Frameworks 6.28 Fixed multiple sources of icons and thumbnails being blurry throughout Kirigami-based KDE software when using a fractional scale factor. (Marco Martin, kirigami MR #2116) Improved the way the Kicker/Kickoff/etc. menus handle the case of multiple applications with the same name being installed. (Christoph Wolk, KDE Bugzilla #516802) Notable bug fixes Plasma 6.6.6 Fixed an issue that could make KWin fail to start at login with some older GPUs and GPU drivers in certain circumstances. (Xaver Hugl, KDE Bugzilla #521333) Fixed a case where Plasma could crash while switching wallpapers in a slideshow. (Fushan Wen, plasma-workspace MR #6753) Fixed an issue in System Monitor that could make items get sorted the wrong way after changing the order or visibility of columns in the data table. (Arjen Hiemstra, KDE Bugzilla #518073) Fixed a regression in the icon displayed by the Networks widget for VLAN networks. (Fabian Druschke, KDE Bugzilla #516709) Plasma 6.7.1 Fixed one of the most common ways that Discover could crash on operating systems using the rpm-ostree packaging technology, such as Fedora Kinoite. (Timothée Ravier, KDE Bugzilla #519672) Fixed a case where KWin could crash while logging into the system if an app asked for an invalid color format. (Shaun Ren, kwin MR #9382) Fixed an issue that could break accented text entry for some keyboard layouts while the Plasma virtual keyboard was enabled. (Kristen McWilliam, KDE plasma-keyboard MR #126) Fixed a regression that made some network details go missing from the Networks widget. (Tobias Fella, KDE Bugzilla #521415) Fixed a regression that made Task Manager badges displaying very large numbers of unread items visually overflow on very thin panels. (Nate Graham, KDE Bugzilla #521562) Fixed a regression that made the Kickoff Application Launcher widget resize itself inappropriately while showing search results right after being added to a panel. (Christoph Wolk, KDE Bugzilla #521383) Fixed a regression that made the Kickoff Application Launcher widget lose its hover effect for category list items, but only when using the non-default "Switch sidebar categories when hovering over them" setting. (Christoph Wolk, KDE Bugzilla #https://bugs.kde.org/show_bug.cgi?id=521558) Fixed two issues with clipboard syncing in remote desktop use cases. (David Redondo, xdg-desktop-portal-kde MR #588 and xdg-desktop-portal-kde MR #589) Frameworks 6.28 Fixed a regression in Kirigami that made the header of the "Manage Panels and Desktops" window overlap with the content. (Marco Martin, KDE Bugzilla #521552) Notable in performance & technical Plasma 6.6.6 Fixed a significant memory leak that was found when using window rules to force window decorations on Firefox in CSD mode. (Xaver Hugl, KDE Bugzilla #518939) Plasma 6.7.1 Turned off the game controller input plugin by default, as some issues were uncovered and it appears to need a bit more time in the metaphorical oven. (Xaver Hugl, kwin MR #9424) How you can help KDE has become important in the world, and your time and contributions have helped us get there. As we grow, we need your support to keep KDE sustainable. Would you like to help put together this weekly report? Introduce yourself in the Matrix room and join the team! Beyond that, you can help KDE by directly getting involved in any other projects. Donating time is actually more impactful than donating money. Each contributor makes a huge difference in KDE — you are not a number or a cog in a machine! You don't have to be a programmer, either; many other opportunities exist. You can also help out by making a donation! This helps cover operational costs, salaries, travel expenses for contributors, and in general just keeps KDE bringing Free Software to the world. To get a new Plasma feature or a bug fix mentioned here Push a commit to the relevant merge request on invent.kde.org.

- [Web Review, Week 2026-25](#) (2026/06/19 06:15)

Let's go for my web review for the week 2026-25. Open Source vs the Invisible Hand Tags: tech, foss, commons, economics Or why modern economics mostly lose the plot when you try to factor Open Source in these simplistic theories.
<https://nesbitt.io/2026/06/18/open-source-vs-the-invisible-hand.html> Recommendations When Using LLM-backed Generative AI Systems for FOSS Contributions Tags: tech, foss, ethics, ai, machine-learning, gpt, copilot A very balanced set of recommendations from the SFC around LLM uses. It's just the beginning and still lacks a bit in details. It's very welcome though and I look forward to their updates.
<https://sfconservancy.org/llm-gen-ai/llm-backed-generative-ai-recommendations.html> Turn your site into a place people can bump into each other Tags: tech, web, blog, community I like this idea. It'd be nice if more websites felt like little town squares.
https://cauenapier.com/blog/townsquare_release/ The world's first trillionaire is a killer Tags: tech, business, politics People excited by accumulating so much wealth (on paper) are clearly showing sociopathic traits...
<https://www.theverge.com/tech/949259/the-worlds-first-trillionaire-is-a-killer> Extinction-level capitalism Tags: tech, politics, ethics, economics, ai, machine-learning, gpt Long, rich, and sourced piece. Or why the current gold rush aims at accelerating wealth accumulation of a few to the expense of everyone else. If the plans work as intended, the outcome won't look good.
<https://matthewbutterick.com/extinction-level-capitalism.html> Dangerous Technology For Americans Only Tags: tech, europe, politics, ai, machine-learning, gpt, foss, commons The latest move by the US government treating LLMs like dangerous weapons tells something about the geopolitical moment. Can we collectively raise to the challenge and build on cooperation instead? It'd be a much better position than assuming governments or big companies will make the right choices for everyone else in isolation. <https://lucumr.pocoo.org/2026/6/13/americans-only/> It Is Trivially Easy to Use Reddit to Manipulate AI Search, Research Suggests Tags: tech, ai, machine-learning, gpt, fake-news, scam, attention-economy What a surprise... It turns out it's very easy to manipulate AI "search". Something which operates of statistical similarity to queries, who knew it could be manipulated. <https://www.404media.co/it-is-trivially-easy-to-use-reddit-to-manipulate-ai-search-research-suggests/> Report: How Developers React to AI-Scented Blog Posts Tags: tech, blog, ai, machine-learning, gpt Badly apparently, looks like it makes for prose people avoid. Now the thing is... with the widespread suspicion, some people might be wrongfully flagged as using LLM to write their posts.
<https://writethatblog.substack.com/p/dev-reaction-to-ai-blog-posts> The Future of the Con Is Already Here, It's Just Not Evenly Distributed Tags: tech, ai, machine-learning, gpt, scam, security What happens when targeted scams become cheap to run? This covers it fairly well, and we need to change our heuristics and trust model. <https://manishearth.github.io/blog/2026/06/17/the-future-of-the-con-is-already-here/> xan: The CSV magician Tags: tech, data, data-visualization, tools, command-line Looks like a nice command line tool for exploring data.
<https://github.com/medialab/xan.gitignore> Isn't the Only Way To Ignore Files in Git Tags: tech, tools, version-control, git Neat little summary on the mechanisms to ignore files in Git. <https://nelson.cloud/.gitignore-isnt-the-only-way-to-ignore-files-in-git/> Boot Naked Linux Tags: tech, linux, embedded A good reminder that Linux based systems can be slimmed down quite a bit. Can come in handy for specific devices.
<https://nick.zoic.org/art/boot-naked-linux/> Why stdx is not on crates.io Tags: tech, rust, supply-chain, security Indeed, skipping the centralized package manager might be better in the long run. <https://kerkour.com/stdx-cratesio> Async Task Locals From Scratch Tags: tech, asynchronous, memory, rust Wondering how those are implemented and the challenges behind them? This is a good exploration of the idea.
<https://wolfgirl.dev/blog/2026-06-16-async-task-locals-from-scratch/> Learn Rust Concurrency By Building a Thread Pool Tags: tech, rust, multithreading This is an excellent exercise to understand a language and its ecosystem better. This time it is about Rust.
<https://blog.sheerluck.dev/posts/learn-rust-concurrency-by-building-a-thread-pool/> Rust Prevents Data Races, Not Race Conditions Tags: tech,

rust, multithreading Sounds obvious to me, but I guess it's worth repeating. It illustrates well the type of issues you still have to deal with in multi-threaded Rust. <https://corrode.dev/blog/rust-prevents-data-races-not-race-conditions/> How memory safety CVEs differ between Rust and C/C++ Tags: tech, security, memory, rust, c++, c Interesting take on why CVEs are reported differently for C/C++ and Rust libraries. The responsibility for API misuse is treated differently because the abilities to express contracts is treated differently.

<https://kobzol.github.io/rust/2026/06/15/how-memory-safety-cves-differ-between-rust-and-c-cpp.html> Memory Management in C++ by Patrice Roy Tags: tech, c++, memory, book Looks like a good book. This review definitely tempts me.

<https://www.sandordargo.com/blog/2026/06/17/memory-managenent-by-patrice-roy> Parsing JSON at compile time with C++26 static reflection Tags: tech, c++, type-systems, compiler, reflection This is really interesting stuff that C++26 brings.

<https://lemire.me/blog/2026/06/14/parsing-json-at-compile-time-with-c26-static-reflection/> Every Frame Perfect Tags: tech, ux, graphics, animation, gui Indeed, the intermediate steps in an animation have to make sense too. Too often we think about the start and end point but miss the quality of the transition. <https://tonsky.me/blog/every-frame-perfect/> Getting Creative with Perlin Noise Fields Tags: tech, graphics, mathematics So many possibilities with Perlin noise! There are some really fascinating textures in the lot.

<https://sighack.com/post/getting-creative-with-perlin-noise-fields> The case for real collaboration Tags: tech, pairing I still wish our craft would practice pair and ensemble programming more. Somehow it's been years and it doesn't really take off unfortunately.

<https://blog.mikebowler.ca/2026/06/15/collaboration/> Nine Questions I Now Ask in Interviews That I Wish I'd Asked Five Years Ago Tags: interviews This is a good list of questions. Never forget you need to assess the company you're applying at. To often candidates forget to do it.

<https://louisedeason.substack.com/p/nine-questions-i-now-ask-in-interviews> When Overwhelmed, Slow Down Tags: health This is sound advice. Especially in this era of chasing productivity at all costs, we tend to not listen to ourselves enough.

<https://nathanpeterson.net/overwhelmed-slow-down/> Bye for now!

- [Ocean Updates - June 2026](#) (2026/06/17 23:15)

Penpot WebGL and Bugfixes Penpot and WebGL Penpot released their WebGL rendered (beta) for all users to test. This brought speed gains in rendering content on page and less lag when navigating. Frame rate was worked on and the overall interface appears faster. Pages load much faster than before, though there are still some rough edges. Overall, this is a gain for Ocean given it is a large design system. This change also brought a sleuth of new bug reports for Penpot, you can see them here:

<https://github.com/penpot/penpot/issues?q=is%3Aissue%20state%3Aopen%20label%3A%22IOP%20render%22> We also experienced a few issues around speed, borders missing, and incomplete rendering. Those were reported promptly. From what I can see in their development page, there is going to be a 16.1 release with a few fixes for the rendering engine that should address the most pressing issues around speed and rendering accuracy. Issue with boolean shapes and Penpot crashes It seemed for some time that the Penpot file for Ocean was crashing constantly. I wasn't sure what the problem was but whenever there was an update to the tokens, the system would record the change and then crash. After reviewing some of the error messages, it seemed that the problem was derived from shapes containing boolean operations. For example, the Date Picker component has a cutout circle (A difference boolean operation) to form the circular shape that highlights the current day. Those components appear left and right of the circular shape. Like half moons. These shapes were not flattened, they retained their boolean operation. This is generally not an issue at all but it appears it was in Penpot. Once I identified the shapes with the boolean properties, I flattened them all to make sure they were no longer a problem, and then boom! The problem was gone. However, one reason for keeping boolean operations in the

components and not flattening is so that other designers can understand how some components are built and also have greater control over their design. When you flatten shapes, this makes it so that you lose the “how” a shape is built. Even though crashing is resolved, it’s important to note that adding, editing, and removing tokens can take a long time. Issue with Input Fields and Badges Set to Static Whenever one changes the base component at the root of the component, it seems that propagation is not always clean and some aspects of the component do not update as they should. Some values tend to go back to being static when you mean for them to use the fill or hug property. It appears that some recalculations do not always yield the right result on the components. They are aware of these problems and are working to address them. This issue happened in some components where the text fields inside a nested component were supposed to have the fill property, but in their wrapped component setup, the base component text field showed static. This made it so that input fields looked longer than they should have inside other bigger components. It took me a minute to update them all but it seems that now the input fields and badges should behave properly. After putting all these fixes in, I uploaded a new version of the Penpot file into gitlab so that others can update. Figma Icon Flattening We still have the issue that we can’t bring our icons to Penpot just yet. As we build more and more UI concepts for Ocean inside Penpot, we need to rely on our new Ocean icons to make sure they convey the right message in context. I went back to the Figma file this week and created a copy of our monochrome icons and started the process of flattening them. This makes it so that it’s just a shape without layers or boolean operations. Just a raw flat svg file. This type of operation is supported in Penpot, though we lose the fine control. My desire is to bring some, if not all, of our monochrome icons into Penpot, even if they are flat, so that we can use them throughout our component UI more effectively. I am still exploring whether this is a good idea overall but this should add a few more images to the Penpot file. Expect the Penpot library to grow. This is an experiment to see how Penpot will handle this influx. App Icons Progress Progress is moving slowly but surely. I went through a round of feedback review as I posted some of the icons into the VDG channel and also asked Nuno Pinheiro for feedback. There are a few more things that need to be done. As shown previously, the work of building colorful icons goes in phases. The current phase is intended to build colorful, “try anything”, higher fidelity icons. In this stage, shapes, colors, radii and other elements are placed on screen to see how they come across. From this stage, one would pick out the best icons and use them as design guidelines for other icons. You begin to restrict your color palette, harmonize shapes, shadows and other elements. This process will continue. My personal goal is to have a robust set of app icons for Academy in September (<https://akademy.kde.org>). Then work with maintainers to get buy-in and feedback for the new designs. Tokens and Plasma System Variables We follow the token naming convention used in Material Design 3. It’s expressed like this: Raw value > Reference Token > System Token Plasma has its own set of variables used for various purposes. For example: SpacingSmall, SpacingMedium, etc. We need to find the way to do Raw value > Reference Token > System Token > SmallSpacing Even though the design system contains more tokens than there are variables in Plasma, we need to do the work of matching design tokens to Plasma. This should yield a workable set of UI changes even though it may not be complete. For those tokens that don’t have an equivalent in Plasma, we need to create an integration and support plan. For those design tokens that have approximations, we need to determine what is best, to accommodate to what we have in Ocean or to make Ocean accommodate to what the system already has. The biggest risk here is that if we change values in the system carelessly or introduce unsupported tokens, the applications using those variables in Plasma could break or work in a strange way. In this new age of Union work, we would parse design tokens via Union, this would produce the graphical interpretation of the token and apply it. Since Union is using CSS to interpret the values, token values and component values can more easily be replicated in Plasma.

- [Qt Contributors Summit 2026: Oslo - 27th to 30th October!](#) (2026/06/17 20:42)

This year Qt Contributor Summit will be held in Oslo during late October. If you are a Qt contributor (developer, forum question answerer, promoter, etc) you should think of attending, it's always a great experience :) More info in the announcement at <https://www.qt.io/blog/qt-contributors-summit-2026-oslo-in-october> If you want to attend do not forget to register (the earlier the better for the organizing team) following the form at https://wiki.qt.io/Qt_Contributors_Summit_2026 Thanks to The Qt Company for organizing this and also to reMarkable for hosting!

- [Amarok 3.3.3 released](#) (2026/06/17 11:45)

The Amarok Development Squad is happy to announce the immediate availability of Amarok 3.3.3, the third bugfix release for Amarok 3.3 "Far Above the Clouds"! 3.3.3 features a number of small bugfixes, e.g. to saving of window and context applet layouts, inhibiting suspend during playback and updating of Wikipedia applet to handle current syntax. Additionally, small code quality improvements and fixes for building the software on different systems with various setups are included. All in all, Amarok 3.3.3 keeps ensuring you can keep enjoying your music in 2026. Changes since 3.3.2 CHANGES: Small code quality improvements and fixes for building on different systems BUGFIXES: Restore functionality for inhibiting system suspend when music is playing Fix restoring main window layout after restart (BR 514963) Fix context applet height saving sometimes not working (BR 514190) Adapt parsing in Wikipedia applet to handle some page syntax changes Don't multiply tracks when dragging from file browser to PUD (BR 521425) Getting Amarok In addition to source code, Amarok is available for installation from many distributions' package repositories, which are likely to get updated to 3.3.3 soon, as well as the flatpak available on flathub. Packager section You can find the tarball package on download.kde.org and it has been signed with Tuomas Nurmi's GPG key.

- [Qt Creator 20 released](#) (2026/06/17 10:18)

Release 20 of the Qt Creator IDE adds support for working with AI coding agents, a Zen Mode that puts your code editor into the focus, support for the GN (Generate Ninja) build system, and many more improvements.

- [Libre Graphics Meeting 2026](#) (2026/06/17 00:00)

This is supposed to be a blog post about Libre Graphics Meeting (LGM) this year, but since I haven't written about the previous ones I went to, I decided to write about those as well. So this post will have bits of 2024 and 2025 in them as well, as far as I remember them, along with the experiences from the 2026 edition. And also some small bits about what's planned for the upcoming ones! I'm not much of a taker of photos, so I don't have much to show in that regard, but I got stories and links. Some of the conference talk links go to YouTube in absence of alternatives. At the time of writing, the talk recordings are still being moved from a Peertube instance that's going away to the Chaos Computer Club's excellent media hosting service. Presumably in the future, you will be able to find all the LGM talks categorized by year under the conferences/lgm category or under events/lgm or somewhere else on media.ccc.de. I've also never had a personal blog, so I guess a brief introduction: I'm Carsten, I develop software. I'm involved in Krita, my main focus there being on Android. I'm also the maintainer of Drawpile. The two programs have plenty in common and so there's a good amount of cross-contamination. With that out of the way, I'll start not at the beginning, but at the now. Next Year: LGM 2027 Next year's Libre Graphics Meeting will be from May 5th to 8th 2027 in the city of Linz, Austria at the Kunstuniversität (art university). That's Austria the country in Europe, not Australia. The talks will probably be streamed again for online viewing. There is also some mulling about LGM 2028 going on already, with the idea of it being hosted in Montreal, Canada being floated about. Since it's still very early in the organization, you probably won't be able to find those dates listed anywhere yet, but you can probably find them in the Matrix channel backscroll. That is also a good place to go if you want to listen in on the meetings and participate. Otherwise, if you want to keep up to date with

the event, you can also follow the Fediverse account. Otherwise, the website also has information, but at the time of writing, it is still in its 2026 state. Two Years Ago: LGM 2024 This was my first Libre Graphics Meeting. It was also the first one in quite a while, it took quite the hit from COVID and the usual yearly cadence it had kept up for over 15 years had lapsed. 2020 and 2021 were online, 2022 and 2023 didn't have any LGMs at all. 2024 was the first year it came back. I don't remember it that way, but looking at the group photos from then, the crowd is definitely smaller than in the subsequent years. The venue for it were the ActivDesign offices, which is probably a comparatively tighter space, making it seem like it was more folks around. I'd taken over the maintenance of Drawpile, having some more time for that kind of thing after quitting my day job. That also involved porting the program to Android, which is how I ended up on the Android side of Krita, and to the web browser using WebAssembly. The latter felt like something that few people had done before and I had stories to tell about it, which is why I gave a talk on the topic. While this is probably my first publicly recorded talk, I've given enough of them before that this wasn't a new experience. However, it was the first one that had a good chunk of less technical people in the crowd, so I tried to have some lighter parts, at least as far as possible in such a techy topic. I'm pretty happy with how it turned out and people seemed quite interested in it, given the amount of questions and "hallway" conversations I had. There was also a talk about Krita's new text tool by Wolthera, the primary author thereof. She couldn't be there in person, so that talk was given remotely. She's also got a blog post series on the matter. Today of course, Krita has this new text tool she was working on! I don't remember every talk and so I don't want to go over each of them, but many of them I still remember fondly and it was nice seeing the people that held them at subsequent LGMs again. The folks from ActivDesign of course, like Elisa, Animtim or Cédric, who were involved in organizing the event. Tom Lechner's talks are always interesting and almost given in a conversational manner. This year's being one of hiking, foggy mountains in the Pacific North-West and default cubes in their natural habitat. It was also the first time I've really seen "live computer art stuff" through Jules Fouchy's Coollab. Well, what that stuff is really called is "generative art", but of course if you search for that nowadays, you just get slop instead. This is not that anyway, it is more about creating living installations created in real time and transforming them, like creating live visuals given an audio feed, manipulatable by a gyroscope from a performer's phone. There was also a bit of a conference talk with AMRO, the Art meets Radical Openness festival in Linz, Austria. Which is where next year's LGM connection comes from and some of the people involved with AMRO are also doing the organization of LGM 2027! You can find all the recorded talks in this playlist, unfortunately the lightning talks are not part of it. Outside of the conference itself, there's also plenty of memories. Rennes was a nice city, my first experience with it being at the train station and desperately trying to find a ticket machine for the bus. After what must have been an hour of walking from the top of the station to the underground metro, I gave up and just got on the bus without a ticket - only to find that you just touch your bank card to a contactless pay terminal. Since the fees are a single, flat price no matter where you're going, that's all you need. There was also much Brittany-typical stoat iconography in the city. After a picnic on the first day, we went out to eat on the second one. We had galettes, which are basically savory buckwheat crêpes. If I recall correctly, GIMP sponsored those for everyone that was there. I also had a bit of a talk with Lasse Fister, the lead organizer from the subsequent LGMs, about his attempts of having software recognized as being "for the public good" in Germany. Which informed that direction with Drawpile to instead focus on the artistic side of things, rather than trying to explain free software to the tax office, which seems like it was the right path because it lead to it having its public goodness recognized without a hitch. I unfortunatly had to leave on the third day, so I missed out on the rest of the meeting, a mistake I didn't repeat again. And while I don't have photos, I still have my shirt, badge and official coffee cup! It's even signed, although that has faded a bit. Nonetheless a precious item. Last Year: LGM 2025 The next LGM was in Nuremberg, Germany, which is quite a lot easier for me to get to. This time it was a larger space at the Karl-Bröger-Zentrum, which had

more room available. Lasse put everything together despite plenty of hurdles and the support structures from one and a half decades of LGMs not being there anymore. During the event Simon Harhues also took on a good amount of organizational tasks, spontaneously if I recall correctly. The tech was done by folks from Chaos-West TV, who did a tremendous job despite the venues at times questionable tech. At times it felt like the recordings of the talks were already cut and online before the speaker stepped off the stage. And as the most important person of the event, Lasse's mother prepared the coffee. Several people that were there at the last LGM were around once more, who were nice to see again. And there were several others that would come back the next year again. Krita also had a physical presence this time with tiar being there, myself not being a member of the core development team yet, although I'd already made some contributions to it. GIMP and Inkscape also had a presence again. I gave a talk again this time around, with the lurid title of 3000% Faster File Saving With Time Travel. It's about a new file format for Drawpile, which turned out to have some very cool properties that I thought could be applicable to other programs as well. By now, a good chunk of the second part of the talk that goes on about future stuff has become a reality. A similar kind of format may be coming for Krita as well, although it will take some experimenting with what's doable in that regard. This year also brings the continuing format of the State of Libre Graphics talk by MABarbe. It features a short overview of what's going on with various projects in the libre graphics world, even if they are not at LGM. If you know any free software projects that aren't represented in this but you think should be, it may be worth letting the project maintainer and/or the LGM groups know about it. Once again, I don't want to retell every single talk since I don't remember all of them, but I'll give a brief survey again. Tom Lechner's talk this time was about waterfalls in Godot, which was another memorable one. Live computer art stuff had more of a presence this time, with Coollab once more, but also OPENRNDR and Processing being around. The talks about PDF were also interesting, one being a lightning talk on CopyPDF and the other on various kinds of flavors of PDF. And, most interestingly for my work, there was a talk on Inkscape's UI, on usability and UX evaluation and a workshop on the latter by Simon Harhues. During that, we went through Krita with someone unfamiliar with the software try to accomplish a task and document the issues. It was pretty eye-opening to see the things that a user tried first and having their thoughts spoken, since even if they did manage to accomplish the task eventually, they took some wrong turns or came to incorrect conclusions. A few of the experiences from this lead to features, such as hitting Escape to deselect in Drawpile. All recorded talks can be found in this playlist. Since the schedule was pretty packed this time and my hostel bed was a small train ride away, I didn't end up going to too many evening events, although there were some concerted lunches. The last day had a big get-together with everyone who still had time, which was a nice end to the event. Looking back at it and comparing it with the LGMs before and after, it really does seem like the meeting is picking up steam again, becoming more organized and eventful. No limited-edition plastic coffee cup this time, but I do have a shirt, a badge and stickers. This Year: LGM 2026 Finally, the event this year. It was in Nuremberg again with the same organizer and tech team, but the venue was Zollhof. The internet was a lot better and the place was better equipped for this kind of conference matter. Coffee was provided without parental guidance this time. There was also a nice community room in the basement, although it didn't end up getting that much use since it was very far away from the conference hall on the very top floor, which is where most of the hallway conversations ended up being. Krita-wise, tiar, Animtim and I were there. GIMP was also around to a degree, although Inkscape had less of a presence this time around. I also didn't have a talk this year, mostly because I didn't really have a single concrete thing to talk about among everything I'd been doing and the new work on the file format I'd already talked about in the last year. The memories here are the freshest, so I'll go into some more detail here on talks where I have something to say. I won't go into every talk, since stuff like fonts, printing or live coding I simply don't have much to say about and I don't want this to just be a summary with no additional insights over just watching the talks. Which you can do via this playlist. Once again, there was a State of Libre

Graphics talk by MABarbe. Drawpile isn't featured this time around, which MABarbe and I blame wholly on Matrix failing to do its job of delivering messages. Jules Fouchy, the primary developer of Coollab, held a talk about something we've chatted about at LGM 2024 already: UI frameworks. My own research and experience was that the choices that could be taken seriously - i.e. runs on Linux, Windows, macOS, Android, iOS/iPadOS and in the browser with possibility of hardware acceleration - were basically Qt, web view (Electron and friends) or an immediate GUI on top of OpenGL (Dear ImGui and friends.) While Coollab has different requirements, he kind of comes to the same conclusion. It is worth the watch. Also on the first day there was a kind of crazy talk by Øyvind Kolås about ctx, a rasterizer, and its vector desktop, which is a whole desktop environment around it. The whole thing is pulling itself up on its own bootstraps, which is kind of wild to see, it is worth watching the talk. After having held a talk on PCB design the last year, Simon Budig held a soldering workshop. Since I have some experience in that regard I ended up joining it to help out a bit, everyone there put together a small board with a few components: resistors, a battery compartment, a switch and blinkenlights on a printed circuit board with the LGM logo on the front. This gets slotted into a 3D-printed case with a magnet inserted into a very tightly fitted recess via a lot of force applied. Turning it on lights up the logo in different colors that shift over time, see the photo below. In the lightning talks on the second day, Krita contributor Animtim held a talk about GCompris, a tool for children with different activities, of which several of the creative ones are quite interesting to play with for adults as well. There was also a talk on palettes appropriate for different color visions by Øyvind. I also talked very briefly about animation in the ORA format to see if there was anyone except Krita folks interested in it, but we didn't have anyone present at the time. Jehan held a talk about GIMP's community, which is worth a watch if you're involved in a free software project. It could probably have been twice as long to go more into depth on the topic, but it gives a good history and some insights on how to handle such a long-running project. In particular, their experience with Google Summer of Code students is very different to Krita's: theirs usually turned into long-time contributors, while ours usually disappeared after they were done. While Krita is sitting out on this year's GSoc for various reasons, it'd probably be worth looking deeper into it next year. I didn't quite wrap my head around it during Jean-Michael Celerier's talk on ossia score, but there was a workshop later where I understood it a bit better. It is a program that lets you orchestrate various kinds of input sources, processing steps and output sinks together with nodes and a timeline. For example, you can read sensor data from a phone, denoise it and then output it into a shader program to make visuals during a live performance. Or you can connect a sound source with a light installation. While I would probably have to use it more to wrap my head around the connection between the timeline and nodes, it is an interesting combination of the two. Speaking of timelines and nodes, MABarbe held a talk about representing a timeline as nodes, which is another way to look at the matter, although this one is a UX design experiment, not something available to try and feel out. Still, it's worth watching the talk in combination with the ossia score one above. Lila Pagoda's talk about putting design students in front of free software is also a good watch. It reminds me quite a lot of the UX experiments we did in 2025 and I would really like to read the students' reviews that were mentioned, but haven't found them publicly released. It is also an insight on technical awareness to a degree, where students never even look at alternatives that may even be better than proprietary solutions. But that's probably a marketing matter more than anything, which I don't understand much of. Tom Lechner held a talk again this year, this time about imposition, which is how you cut, fold and arrange papers to produce their eventual "bound" form. It's not something I've ever thought about, so actually seeing it in action is fascinating. Right afterwards, Alexander Lehmann's talk about their Frag den Staat game is in a similar direction, being taken along with the experience of making that game. And as the final talk, flabbet from PixiEditor held a talk about their node-driven editor. Particularly interesting is the use of a node graph for their brush editor, since it makes the flow of information more apparent than the "fixed-function pipeline" in other brush engines. Whereas those tend to accumulate a lot of special knobs

and switches, having a node editor allows for easier stacking of primitives. A particularly interesting part thereof being the ability to have user input nodes, which means that an artist that just wants to tweak the brush a bit doesn't need to dive into the node graph, but can instead just adjust some sliders and curves on the frontend. The conference ended with a get-together outside this time, which was a bit less comfortable due to the lack of seating opportunities in the hills of the old city. Still, we talked about nodes, crash reports and optimization, some of which has conjured ideas on implementing them into Drawpile and/or Krita. Although they haven't yet manifested due to being busy on other matters in that regard. So for the spoils this year. I got a shirt again, badge, stickers and the aforementioned soldering project. And since I already looked at next year's LGM above, that concludes this post! If it's reachable for you, consider coming as well, the attendance is free. Otherwise, save the date to watch the streams and participate from online!

- [Week 3: Reviewer Feedback and Fixes](#) (2026/06/16 16:18)

This is a weekly update from my Google Summer of Code 2026 project with KDE, improving effect widgets in Kdenlive, a free and open source video editor. This week was driven entirely by reviewer feedback on MR !887; the draft MR for the Curves Widget. Point snapping instead of rejectionWhen two curve control points are placed too close on the x-axis, avfilter/MLT crashes with Key point coordinates are too close or not strictly increasing. The original guard in AssetParameterModel::internalSetParameter silently rejected the update, the user moved a point but nothing happened visibly. Reviewer Bernd Jordan flagged this as confusing UX. The fix: instead of rejecting, snap the offending point so it maintains the minimum safe x-distance (~0.00266) from its neighbor. The curve is always valid, always sent to MLT, and the user sees immediate feedback. Removing the 5 point limit for avfilter.curvesThe curve editor had a maximum of 5 control points, inherited from frei0r.curves. JB pointed out there is no reason to keep this limit for avfilter.curves, it is a frei0r-specific constraint. The fix: setMaxPoints is now only called for frei0r.curves. The avfilter.curves widget has no upper limit on control points. Both fixes are in MR !887. JB also noted the null placeholder approach in m_widgets for secondary av_curve params is not ideal long term, waiting on his direction before touching that.

- [Week 3 — Bug Fixes and Import/Export Feature](#) (2026/06/16 08:22)

This week, I worked on two things: a bug fix that was created during the ActionCollection port and starting the import/export feature. Bug fix: Lock/Unlock action text: A bug was created after !29 was merged. When a wallet was locked, the placeholder message displayed a "Lock" button instead of "Unlock". The issue came from ActionData using static text, while the lock action needs to reflect the wallet's current state. The fix was easy — switch between the lock and unlock ActionData dynamically based on the locked property. AC.ActionCollection.action: locked ? "unlock" : "lock" This was submitted as !31 and merged. Configure Shortcuts menu item: I added a "Configure Shortcuts..." menu item to the globalDrawer(!32), using AC.StandardActionData.KeyBindings to attach to the action automatically created by ActionCollectionManager. Import/Export feature: I started implementing the import/export feature. The format is KWalletManager-compatible XML, so users can migrate between KWalletManager and KeepSecret. The implementation adds: -ImportExportManager C++ class with exportToFile() and importFromFile() methods. -Export... and Import... menu items in the globalDrawer. -File picker dialogs using QtQuick.Dialogs. The merge request !33 is open and under review.

- [Oxygen 6.7 is here: a breath of fresh air for KDE's classic theme](#) (2026/06/16 06:49)

Logo by: Nuno Pinheiro The year started off bleak. As I was gallivanting through KDE themes at hand, I decided to stick with the Oxygen one. It didn't take long to notice that this old theme, once the default in the KDE 4 era, wasn't looking its best. A slew of little bugs had accumulated,.....

Continue Reading →

- **Plasma 6.7** (2026/06/16 00:00)

Plasma 6.7 brings powerful new features to KDE's classic desktop, and refines its user experience to new levels. Enjoy thoughtfully-designed improvements that solve real problems, a sneak peek of future theming bliss, better performance, and more. Read on to learn all about it! A script element has been removed to ensure Planet works properly. Please find it in the original post. A script element has been removed to ensure Planet works properly. Please find it in the original post. Highlights Per-screen virtual desktops After 21 years, hopefully it will have been worth the wait... per-screen virtual desktops have finally arrived! Test your microphone volume No more being told your audio is too loud or quiet, or wondering if it's your microphone or an app that's misbehaving. Press-and-hold for special characters With Plasma's virtual keyboard enabled, press and hold a key to type special characters related to it. Widgets Light/dark mode toggle In Plasma 6.6, we introduced the ability to create your own Global Themes, and switch between them as part of the day-night cycle. Plasma 6.7 puts you even more in control with a quick toggle to switch between light and dark Global Themes immediately. Vietnamese lunar calendar Plasma 6.7 adds the Vietnamese lunar calendar to its available set of non-Gregorian calendars. Now you can keep up with dates and events using both calendar systems with more ease than ever! "Background Apps" in the System Tray The System Tray is already the place for monitoring apps running in the background using traditional methods. Plasma 6.7 extends this support by also showing apps using the newer "Background Apps" system, which is more common for apps using the Flatpak package format. Printing Plasma's printing support continues to mature. Now the System Tray icon for printers shows the number of active jobs in a little badge. On the enterprise side, Plasma 6.7 makes it substantially easier to connect to shared printers on Windows networks, and a new print queue management tool offers more power than ever before to those administering multiple frequently-used printers — while remaining simple and comprehensible enough for home use, too. Usability Plasma 6.7 puts a particular focus on refining the desktop experience to make common tasks faster than ever. Here's just a small assortment: Faster virtual desktop switching in the Overview When you press Meta + W to open the Overview screen, Plasma 6.7 lets you switch virtual desktops simply by scrolling or pressing the Page Up or Page Down keys. Drag and drop to add and remove favorites It's now easier than ever to add or remove favorite apps in the Application Launcher, Application Menu, and Application Dashboard widgets: just drag and drop! More intuitive software management The Discover software center now features a more obvious "Install" button and redesigned app cards with more information, plus a more intuitive sorting order. In addition, installed software is now grouped by type, making it easier than ever to find what you're looking for. Easier to compare different time zones How much earlier is it in Denver if you're in Berlin? No longer will a question like this require mental gymnastics; Plasma 6.7 makes it easy to see how many hours earlier or later it is in each time zone shown in the Digital Clock widget compared to your local time zone. "Type-ahead" on the desktop For power users, the desktop layout now supports switching to "type-ahead" mode for quickly selecting files when typing on the desktop. You can change this back to activating KRunner if you prefer by Right-clicking on desktop and selecting Desktop and Wallpaper > Icons > Typing on the desktop from the pop-up menu. Theming and styling: Something old, something new Oxygen In preparation for KDE's 30th anniversary later this year, we've brought the Oxygen theme — used by default in KDE 4 — back up to par with the current Breeze theme. Air, the Plasma Style providing a light version of Oxygen, has also returned, and we've reworked our Global Themes to feature light, dark, and twilight versions. Both Air and Oxygen benefit from a multitude of fixes, improvements, and restorations, including support for adaptive opacity, different panel positions, and more. The classic KDE 4 Air and Horos wallpapers also make a return. Breeze The default Breeze theme also continues to evolve, now featuring rounded list and grid view highlights for many applications, and visible click effects for menu items. Plasma's Notifications have been given a fancier animation: they now slide in from the nearest screen edge. This makes them more visually prominent and

less likely to be missed, while at the same time avoiding being too attention-grabbing. Union Though Plasma's theming capabilities are second to none, for years creating those themes has been a difficult and fragmented experience. Plasma's all-new Union theming system intends to streamline theming by making it possible to style Plasma, QtQuick apps, and QtWidgets apps all with a single set of easy-to-write CSS — the world's most popular and documented open standard for this purpose. Plasma 6.7 marks the inaugural release of Union, here with a tech preview of the QtQuick style. Union is under heavy development, and hence off by default. To test it out, make sure the union package is installed (name may differ depending on your OS), and select "Union" in System Settings > Colors & Themes > Application Style. Applications must be restarted in order to use Union, and are expected to look nearly identical to Breeze. If you've found a Union-specific issue, please report it here. You can find out more about the current state and future plans for Union on its developer's blog. Colors, graphics, and performance Plasma has always offered a superb graphical experience, and 6.7 piles on further improvements. No longer do you have to choose between color management using an ICC profile or enjoying HDR content; now both are possible at the same time! In addition, you can now control whether the colors on the screen turn redder at extremely low brightness levels on many AMD laptops. Finally, the team put a great deal of work into optimization, resulting in better performance and lower power usage for CPU-rendered apps, many full-screen windows, and integrated Intel GPUs. But that's not all... Plasma 6.7 has even more to offer, including: The ability to duplicate network connections, so you can quickly create several with only small differences between them — which can be useful in an enterprise or school setting An option to sync mouse and stylus pointers An option to set and change preferred calendar app An option to assign keyboard shortcuts for toggling the global "push-to-talk" microphone mute mode and clearing the notification history, making these actions faster to trigger An easy way of selecting mixed skin tone emojis in Emoji Selector app A System Monitor app that respects your preference regarding storage units (e.g. "GiB" vs "GB") A Global Menu widget that shows menus for apps on other screens by default A "Recent Locations" item in the Application Menu widget Customizable sorting and grouping in the Window List widget Support for many more Wayland protocols and portals To see a high-level overview of all user-facing changes, see the Plasma 6.7 wiki page. To see the full list of all changes, read the complete changelog. In memory of Eric Laffoon Plasma 6.7 is dedicated to Eric Laffoon, who passed away in May. Eric was a longtime supporter of KDE going back decades, offering up his time, money, bug reports, and advocacy. Eric is sorely missed, and we dedicate this release to him.

- [New Kirigami-Addon's Onboarding Module](#) (2026/06/15 16:19)

Even well-designed applications can become difficult to understand when their main workflows involve multiple controls, pages, or unfamiliar interaction patterns. Expecting users to discover these features on their own may lead to frustration, abandoned tasks, and additional support requests. Kirigami-Addons' onboarding module addresses this problem by presenting contextual guidance directly in the user interface, highlighting relevant elements and explaining them when they matter. When implemented through reusable and non-intrusive components, it can improve the user experience without requiring substantial changes to the existing application code. Kirigami-Addons' onboarding module provides a complete set of features for defining and running contextual walkthroughs in Kirigami applications: Declarative steps: attach guidance text directly to existing QML elements without restructuring the interface. Source areas: define the UI region covered by a walkthrough and containing its onboarding steps. Automatic sequencing: visit steps according to their declaration order. Walkthrough groups: create multiple onboarding flows, such as basic and advanced tutorials, over the same interface. Conditional steps: disable individual steps dynamically; disabled or destroyed elements are skipped automatically. Visual highlighting: outline the current element while dimming and blurring the surrounding interface. Configurable appearance: adjust the highlight padding and the intensity and quality of the background blur. Adaptive tooltips: position

guidance above, below, or within the available space around the highlighted element. Navigation controls: move forward or backward, or cancel the walkthrough, using buttons, keyboard shortcuts, or QML methods. Lifecycle hooks: prepare or restore the interface before a walkthrough or individual step is displayed. Dynamic interface support: navigate between pages, tabs, views, dialogs, or other containers before highlighting their elements. Custom content: add application-specific metadata and present images, animations, videos, or other QML content inside the onboarding tooltip. Responsive geometry tracking: keep the highlight synchronized when elements move, resize, or change parents' positions. State monitoring: observe the active walkthrough, source, current step, guidance text, geometry, and navigation availability. Accessible navigation: provide accessible button labels, keyboard operation, right-to-left layout support, and platform-consistent styling.

Adding Onboarding to Your Kirigami Application

This section explains how you can add onboarding features to your Kirigami application.

Basic onboarding declaration

Basic usage of Kirigami-Addons' onboarding module

After importing the onboarding module, mark a common ancestor of the UI elements with `Onboarding.isSource`. This source defines the area where the module creates its overlay, applies the background effect, and searches for onboarding steps. Steps are declared directly on descendant elements using attached properties. Here, `Onboarding.texts` associates guidance with the Create project button, while `Onboarding.start()` begins the walkthrough. Steps are presented in declaration order, allowing onboarding to be added with minimal changes to the existing QML structure. You can use `Onboarding` attached properties to highlight/explain individual controls, layouts/groups, and even sub-controls.

Defining multiple onboarding workflows

Defining multiple onboarding workflows

The `Onboarding.sourceGroups` property declares the workflows available from a source area. The empty string represents the default workflow started by `Onboarding.start()`, while named groups, such as `advanced`, are started by passing their name to `Onboarding.start()`. Each step uses `Onboarding.groups` to select the workflows in which it participates. The entries in `Onboarding.texts` correspond by position to those groups, allowing the same UI element to provide different guidance in each workflow. In this example, `Create project` belongs to both workflows, while `Enable version control` appears only in the advanced one.

Extending onboarding tooltips with additional data

Using additional custom data in onboarding tooltips

The `Onboarding.additionalData` property associates application-specific metadata with a step. In this example, the metadata provides an animated WebP and its caption. A custom component reads the current step through `Onboarding.currentItem.additionalData` and displays this content using `AnimatedImage` and `Label`. Finally, binding the component to `Onboarding.additionalDataComponent` inserts it into the default tooltip between the guidance text and navigation controls, allowing applications to enrich onboarding steps without replacing the complete tooltip implementation.

Examples

This is the example tutorial included in the Kirigami Addons repository. It demonstrates how the onboarding module can highlight individual controls, groups of controls such as layouts, and even sub-controls. It also shows how to attach additional data to onboarding steps, such as a video and its caption, and how to define multiple onboarding workflows within the same application. And this is the onboarding module in `Minuet`: a short guided walkthrough that highlights the most important parts of the exercise screen, explains what each control does, and helps new users understand the flow without leaving the application. The onboarding module provides a declarative and non-intrusive way to introduce users to important controls and workflows in Kirigami applications. With support for multiple workflows, dynamic interfaces, custom tooltip content, lifecycle hooks, and configurable visual effects, it can address both simple guidance and more advanced onboarding scenarios.

A possible direction for future work is allowing each onboarding step to wait for a specific user interaction before continuing. Instead of relying only on the Next button, a step could advance after the user clicks a highlighted control, changes a value, or completes an expected action. This would extend the module beyond explanatory walkthroughs and enable fully guided, interactive tutorials in which users learn application workflows by performing the actual operations.

Give the module a try in your application and

share your experience, suggestions, and use cases. Feedback is especially welcome while these APIs are still being refined for an upcoming Kirigami Addons release.

- [Introducing pkgcli: A nicer command-line interface for PackageKit](#) (2026/06/14 06:22)

For almost two decades, the PackageKit package management abstraction layer has shipped with pkcon as its command-line client. pkcon does its job, but it was always kind of a “testing” front-end for the PackageKit daemon rather than a tool designed for everyday use. The focus has instead been on the GUI tools, automatic system updates, GUI application managers and other front-ends. Its command names mirror the D-Bus API almost one-to-one (get-details, get-updates, get-depends), output is very plain, and there is no machine-readable mode for scripting. Most importantly though, there has been no development on it at all for almost a decade, so pkcon was stuck in its rudimentary state from that era. Since a lot of changes will be coming to PackageKit, and testing the daemon and working with it from the command-line was not very pleasant anymore in 2025/2026, I decided to modernize the tool as part of my work as fellow for the Sovereign Tech Agency last year. pkgcli is the new command-line client for PackageKit. It is built from the ground up to be pleasant to use interactively and easy to drive from scripts. Why a new tool? Of course, instead of introducing a new tool, I could have just expanded pkcon instead. The problem with that approach is that the pkcon utility has been around for so long and its command-line API had ossified so much, that rather than changing it and potentially breaking a lot of scripts relying on its quirks, I decided to introduce a new tool instead. pkcon can still be optionally compiled for people who need it in their scripts and workflows. The goals for pkgcli, and the features it now has are: Human-friendly command names. Verbs that read the way you’d describe the task, instead of mirroring the D-Bus API 1:1: show, search, list-updates, what-provides, instead of get-details and friends. Readable, colored output by default (still respecting NO_COLOR and degrading gracefully). A real scripting mode. A global --json flag emits JSONL instead of fully human-readable output when possible, to make it easier to use the tool for scripting purposes. Sensible defaults. A few defaults have been changed, such as the metadata cache-age, or automatic cleanup of unused dependencies being enabled by default. This is more in line with current defaults by other tools and frontends. We also print package information in a slightly different, more readable way. Better handling of internationalized text. Text should now align properly in the terminal window, and we should no longer have completely chaotic text output on non-English locales (especially Chinese/Japanese). Why not pkgctl? Originally, this tool was called pkgctl, to match other common cross-distro tool names. However, that name was already taken by an Arch-specific distro development tool. When this issue was raised, we decided to just rename our tool to pkgcli with the next release, to avoid the name clash on Arch Linux. Examples! Here are some examples on how to use the new tool (some of which include the abridged output pkgcli prints). Search for anything containing the string “editor” in name or description, then look at the details of one result: `$ pkgcli search editor` Querying [REDACTED] 100% ace-of-penguins 1.5~rc2-7.amd64 [debian-testing-main] acorn-fdisk 3.0.6-14.amd64 [debian-testing-main] ardour 1:9.2.0+ds-1.amd64 [debian-testing-main] audacity 3.7.7+dfsg-1.amd64 [manual:debian-testing-main] audacity-data 3.7.7+dfsg-1.all [auto:debian-testing-main] augeas-tools 1.14.1-1.1.amd64 [debian-testing-main] emacs 1:30.2+1-3.all [debian-testing-main] gedit 48.1-9+b1.amd64 [debian-testing-main] gedit-common 48.1-9.all [debian-testing-main] gedit-dev 48.1-9+b1.amd64 [debian-testing-main] [...] `$ pkgcli show nano` Package: nano Version: 9.0-1 Summary: small, friendly text editor inspired by Pico Description: GNU nano is an easy-to-use text editor originally designed as a replacement for Pico, the ncurses-based editor from the non-free mailer package Pine. [...] URL: <https://www.nano-editor.org/> Group: publishing Installed Size: 2.9 MB Download Size: 646.0 KB Search only within package names rather than descriptions: `$ pkgcli search name python3` Check for updates. refresh updates the metadata, then list-updates reports what’s available: `$ pkgcli refresh && pkgcli list-updates`

Loading cache [REDACTED] 100% ▲ cme 1.048-1.all [debian-testing-main] ▲ gir1.2-gdm-1.0 50.1-2.amd64 [debian-testing-main] ▲ imagemagick 8:7.1.2.24+dfsg1-1.amd64 [debian-testing-main] ▲ imagemagick-7-common 8:7.1.2.24+dfsg1-1.all [debian-testing-main] ▲ imagemagick-7.q16 8:7.1.2.24+dfsg1-1.amd64 [debian-testing-main] ▲ libdrm1 0.22.0.amd64 [debian-testing-main] ▲ libfftw3-bin 3.3.11-1.amd64 [debian-testing-main] ▲ libfftw3-dev 3.3.11-1.amd64 [debian-testing-main]

Explore relationships between packages: `$ pkgcli list-depends inkscape # list what inkscape depends on` `$ pkgcli list-requiring libappstream5 # list what requires libappstream5` Find the package that provides a capability, here the AV1 GStreamer decoder: `$ pkgcli what-provides "gstreamer1(decoder-video/x-av1)"` ✓ `gstreamer1.0-plugins-bad 1.28.3-1.amd64 [auto:debian-testing-main]` You can also have JSON output for most commands! Attach `--json` to any query and pipe the result straight into `jq`. Each line is a self-contained JSON object: `$ pkgcli --json list-updates | jq -r '.name'` `cme gir1.2-gdm-1.0 imagemagick imagemagick-7-common imagemagick-7.q16 libdrm1 libfftw3-bin libfftw3-dev libfftw3-double3` Try it `pkgcli` is built by default alongside the rest of PackageKit since PackageKit 1.3.4. If your distribution ships a recent enough PackageKit, it should already be on your PATH. You can read its man page `man pkgcli` for more information. Feedback, bug reports, and patches are very welcome.

- [KDE Android News \(June 2026\)](#) (2026/06/13 06:45)

Quite a few things have happened around the Android platform support for KDE applications in recent months, so high time for another update on that. Qt 6.11 As already mentioned previously, we have updated the Qt version to 6.11. That has the unfortunate consequence of losing support for Android 8 and older. Due to that we also removed the ARM32 builds, as devices running Android 9 or higher are very likely capable of using ARM64 builds anyway, cutting down the CI cost by a third. The previously often annoying interactions between input focus and the virtual keyboard seems to have improved somewhat with Qt 6.11. Changes in how the back key/gesture is handled however also caused a few regressions, like a double page pop in more deeply nested applications (Kirigami MR 2100). SafeArea support While the previous focus of dealing with “safe” screen areas (ie. parts of the application window not being covered by screen cutouts or system controls like the Android status and navigation bars) had been on not breaking horribly due to Android’s changed default behavior we have meanwhile been working on polishing this to actually look decent. Fixed misplaced drawer handles (Kirigami MR 2102). Fixed NeoChat’s chat bar being placed behind the bottom navigation bar (NeoChat MR 2893). Fixed a double bottom margin in NeoChat (Kirigami MR 2107). Fixed floating buttons in Itinerary getting misplaced (Kirigami MR 2105). Fixed Alligator global drawer expanding into the status bar (Alligator MR 172). Fixed runtime warnings in Tokodon when computing safe margins (Kirigami MR 2110). If you spot places where this still doesn’t work correctly, let us know in the `#kde-android` Matrix channel! Notifications There’s also a number of improvements and extensions for notification handling: Fixed interactions with notification actions or the notification itself having no effect (MR 200). Implemented confirming inline replies (MR 202). This fixes the notification showing a spinner animation indefinitely after submitting an inline reply. Improved icon handling. We can now properly distinguish between what Android calls the “small” and “big” notification icons, a symbolic application icon and an image of e.g. a chat avatar. This will require small changes to applications to get the best result, in most cases providing a symbolic application icon should be enough already (MR 201). A new API for cross-platform notification configuration is in review (MR 203). This will also benefit Flatpak applications. NeoChat notification with avatar images. Safe JNI usage Interaction between our C++ code and Android’s Java platform APIs happens via the so-called Java Native Interface (JNI). That’s a rather low-level C interface with little to no type safety and the need for error-prone handwritten arcane signature strings. Already back in the Qt 5 era I had therefore written a few helpers for a more type-safe use of this in `KAndroidExtras`. Much of this functionality is meanwhile available in

Qt 6 in a very similar fashion, with the JNI array support being the latest addition in Qt 6.9. Compared to raw JNI use this is already a massive improvement, see e.g. MR 204 making use of this in KNotifications. It avoids practically all hand-written JNI signatures as well as much of the manual type conversion. One part is still missing though, type-safe wrappers so that function arguments and property types are checked at compile-time. That has been extracted and rebased on top of Qt's JNI code in KjniExtras now. It's only 10% of the code, with most of the complex template magic gone. As a small downside we are unfortunately losing the ability to test JNI code on Linux with this, as the old approach provided a mock implementation when not building for Android.

Calendar access With the type-safe JNI wrappers small enough now to be copied as a single header file, this finally unblocked the move of the Android platform calendar backend from Itinerary upstream to KCalendarCore (MR 242). Together with the calendar runtime permission API already in Qt, this should make e.g. an "add to calendar" feature for Kongress or KTrip easy to add now.

File dialogs and remote files An often reported issue against several of our applications is that they seemingly don't do anything when opening a file via the platform file dialog. This happens for files on a cloud share, which the Qt Quick file dialog silently discards due to not being local files, making this look like as if the user has canceled the dialog to the application. Therefore just nothing happens and no error message is shown either. Interestingly enough, that problem also happens on KDE Plasma, where the native file dialog also can select remote files, it's just much less common there. But since we can tell the Plasma file dialog to only allow selecting local files, this one is easy to fix (CR 742273, available in Qt 6.12). On Android we don't have that option, nor would that be really satisfying anyway, opening files from a cloud share is a very valid usecase. Therefore there's now also a proposed change to the Qt Quick file dialog to optionally allow selecting arbitrary URLs, similar to what its Qt Widgets counter-part already offers (CR 743681). This won't automatically fix the problem, but it would at least give applications a chance to do something about this.

Locale-aware sorting Something fairly basic that Qt on Android so far didn't do properly (at least when not bundling it with multi-10MB worth of ICU libraries) was locale-aware sorting. When using English you might not notice that, but in many other languages this results in weird and confusing lists. In German for example the letter "Ä" gets basically treated like "A" for sorting, while so far it ended up after "Z" on Android. There's now a proposed Qt patch (CR 741548) implementing a QCollator backend for Android using platform infrastructure. This uses Android's native ICU flavor when available and otherwise falls back to the less efficient and less featureful Java API. Austria is no longer last in the list when using a German locale.

Crash reporting While we have automatic crash reporting on Linux since some time (see e.g. Harald's LAS talk for more details), crashes on Android were not handled at all by our applications so far. Based on discussions at the Graz Sprint in April this has now changed, KCrash can now detect a previous crash when starting an application, and offers to submit an automatic crash report to KDE's Sentry instance. Those reports have been very helpful on Linux already, providing very important information about issues and allowing to prioritize those with the most impact, but it's nevertheless crucial we don't submit anything without user consent. Crash report dialog on next application start. The bulk of the implementation is in KCrash MR 101, a few changes are necessary for integrating this into applications as well though. In your build.gradle, add sentry-android-core as a dependency: `dependencies { ... implementation 'io.sentry:sentry-android-core:8.43.0' }` In your AndroidManifest.xml, add a meta-data entry configuring the Sentry DSN for your application: `<application ...> <meta-data android:name="io.sentry.dsn" android:value="https://<token>@crash-reports.kde.org/<app-id>"/> </application>` And finally, remove the build system and preprocessor conditions excluding KCrash use on Android. Note that verifying this part is crucial, without KCrash you'll get the aggressive default behavior of Sentry, uploading without user consent.

Inhibition Earlier this year KGuiAddons got a new API for inhibiting system actions such as locking the screen. The obvious usecase for this is a video player, but e.g. Itinerary uses this as well for ensuring your screen stays on while showing a barcode to be scanned at a ticket check. Android platform support for this has also been added

(MR 203), allowing the removal of corresponding code in applications. Outlook While all of that is good progress, things are likely about to change. Later this year Google is planning to roll out measures making it significantly harder to provide and install applications on Android. For more information check out Keep Android Open, a campaign supported by the KDE e.V. among many other organizations. Regardless of how this will eventually materialize, the direction is clear, Android isn't going to be a viable long-term platform for FOSS software, not even in its Google-free form. I have mostly considered it a stop-gap solution until Linux on the phone is ready anyway, so this is another reason to increase the effort into that direction.

- [This Week in Plasma: 6.7 is Very Close!](#) (2026/06/13 00:00)

Welcome to a new issue of This Week in Plasma! This week the Plasma team put the finishing touches on Plasma 6.7 with another big push on bug fixing. It's looking really good for release next Tuesday! As a result, some feature work and UI polishing started to trickle in for Plasma 6.8. Check it out: Notable new features Plasma 6.8 Plasma Browser Integration now supports the Flatpak version of Microsoft Edge. (Conley Dawson, KDE Bugzilla #521109) Notable UI improvements Plasma 6.6 Window actions that involve the mouse wheel no longer respect your "natural scrolling" preference; we reasoned that in this situation, up should always mean up and down should always mean down. (Vlad Zahorodnii and David Edmundson, KDE Bugzilla #442789) Plasma 6.7 If you authorize an app to be able to remote-control the system without asking for permission first (for example, a remote desktop app), when it does so, now Plasma shows a notification that it's happening. (David Redondo, xdg-desktop-portal-kde MR #571) When navigating by using the number pad keys to move the pointer, pressing multiple keys now moves the pointer in a direction halfway between them. (Vlad Zahorodnii, KDE Bugzilla #486520) KRunner-powered searches now suppress results from the "Global Shortcuts" provider when there are better results from other ones, which makes the search results more relevant for common searches. (Oliver Beard, KDE Bugzilla #3710) Changed the automatic day/night theme switcher to switch halfway between the start and end of dawn or dusk, rather than at the end. (Vlad Zahorodnii, Bugzilla #511973) Added kde-shader-wallpaper to the list of allowed wallpaper plugins in Plasma Login Manager. (y4m y4m, plasma-login-manager MR #141) Plasma 6.8 Improved Plasma's ability to detect dark GTK 2 themes and apply a matching icon theme, which should substantially reduce cases of illegible icons in old GTK 2 apps when using a dark color theme. (Luan Oliveira, kde-gtk-config MR #144) The top edges of non-maximized Breeze-themed windows now have as much extra draggable area as the bottom and side edges already do. (Sergey Katunin, KDE Bugzilla #504225) Made the portal-based permission dialogs more consistent in their presentation and wording. (Nate Graham, xdg-desktop-portal-kde MR #575) Frameworks 6.28 You can now use the Meta key on its own to trigger KWin's Overview screen. (Vlad Zahorodnii, KDE Bugzilla #518302) Improved the alignment of thumbnail previews in open/save dialogs. (John Doe, frameworks-kio MR #2249) Notable bug fixes Plasma 6.6.6 Fixed an issue that made KWin accidentally leak the CAP_SYS_NICE capability to child processes it launched, which would break the Bubblewrap sandboxing system when applied to those processes. (Vlad Zahorodnii, KDE Bugzilla #521013) Fixed a bug that could make Plasma crash when changing the monitor layout during the login process. (Marco Martin, KDE Bugzilla #510477) Fixed an issue that could make the "Today" button on the Digital Clock widget's calendar popup highlight the wrong day when in a time zone later than UTC and the local time was before midnight in UTC time. (Fushan Wen, KDE Bugzilla #521114) The feature that lets apps screencast without asking permission no longer requires the screen setup to be identical to what it was when the permission was granted. (David Redondo, KDE Bugzilla #519122) SVG-based wallpapers are now able to fully participate in the dark/light wallpaper switching feature. (David W., KDE Bugzilla #519168) Fixed a bug that prevented explanatory text showing up as expected in the generic "[app] wants to do [thing]" portal dialog. (David Redondo, xdg-desktop-portal-kde MR #584) Plasma 6.7 Fixed a very odd issue that would break the Icons-Only Task Manager widget if

you created a file named metadata.desktop right inside your home folder. (Christoph Wolk, KDE Bugzilla #521247) If you somehow manage to delete a multi-activity setup's active activity, Plasma now switches you to the next one instead of crashing and leaving you with a broken desktop. (Angel Parra, KDE Bugzilla #521124) Fixed a case where Discover could crash while processing changes to distro packages. (Aleix Pol Gonzalez, discover MR #1335) Fixed a case where Plasma could crash when waking from sleep after monitors were added or removed during sleep. (David Edmundson, KDE Bugzilla #https://bugs.kde.org/show_bug.cgi?id=521078) Fixed a case where System Monitor would crash when quit while the column configuration dialog was open. (Nicolas Fella, KDE Bugzilla #491000) Worked around an issue in GTK 4 that could make selected text in some GTK 4 apps become irritatingly de-selected. (Vlad Zahorodnii, KDE Bugzilla #517573) Fixed a weird regression that could make the pointer inappropriately display the "app is launching" animation when minimizing windows using the Plastik window decoration style. (Vlad Zahorodnii, KDE Bugzilla #516264) Fixed a regression that made non-random wallpaper slideshows start over from the first one at every login, instead of remembering the last-seen wallpaper. (Fushan Wen, KDE Bugzilla #512559) Fixed a regression that made System Monitor's Process Table view display unformatted data. (Arjen Hiemstra, libksysguard MR #476) Fixed a very weird issue that could make KWin get confused and stop running animations and animated effects properly after opening an app that creates an invisible window. (Vlad Zahorodnii, KDE Bugzilla #519789) Fixed an issue that could sometimes make the Weather Report widget endlessly reload after waking the system from sleep. (Ce Sun, KDE Bugzilla #517280) Made System Monitor graphs' axis labels show fractional values when needed. (Tobias Fella, KDE Bugzilla #521041) Made it possible to un-favorite apps that are marked as favorites in Kickoff/Kicker/etc. even if they contain some weird character combinations. (Christoph Wolk, KDE Bugzilla #520894) Fixed an issue that could make the notification icon in the System Tray get stuck in a half-rotated state if the animation got interrupted for any reason. (Kai Uwe Broulik, KDE Bugzilla #458156) Fixed an issue that made a translated label turn into an English label after changing and saving settings on System Settings' Screen Locking page. (Sergey Katunin, KDE Bugzilla #521293) Made the Color Picker widget not visually overflow when placed inside a Grouping widget. (Tobias Fella, KDE Bugzilla #517052) Frameworks 6.28 Fixed a weird issue that could make Plasma freeze if you created a .desktop file, set its icon to be a local AVIF image, and put it on the desktop. (Akseli Lahtinen, KDE Bugzilla #521200) Qt 6.11.2 Fixed an apparently fairly common way that Plasma could crash when fetching album art for remote media (for example, on YouTube) shown in Media Player widgets. (Mårten Nordheim, KDE Bugzilla #505490) How you can help KDE has become important in the world, and your time and contributions have helped us get there. As we grow, we need your support to keep KDE sustainable. Would you like to help put together this weekly report? Introduce yourself in the Matrix room and join the team! Beyond that, you can help KDE by directly getting involved in any other projects. Donating time is actually more impactful than donating money. Each contributor makes a huge difference in KDE — you are not a number or a cog in a machine! You don't have to be a programmer, either; many other opportunities exist. You can also help out by making a donation! This helps cover operational costs, salaries, travel expenses for contributors, and in general just keeps KDE bringing Free Software to the world. To get a new Plasma feature or a bug fix mentioned here Push a commit to the relevant merge request on invent.kde.org.

- [Web Review, Week 2026-24](#) (2026/06/12 12:30)

Let's go for my web review for the week 2026-24. Total Reciprocity Public License Tags: tech, foss, licensing, copyright More an experiment than something I'd recommend for real. Still it shows there's a gap we need to close in the licenses available. Let's hope the OSI and the FSF will do strong moves toward closing this gap. <https://trplfoundation.org/> Forms of Open Source Government Tags: tech, foss, governance, satire In part useful, in part satire I think. Still it gives a good idea of various governance models in FOSS communities.

<https://nesbitt.io/2026/06/09/forms-of-open-source-government.html> Retro-Tech Parenting Tags: tech, culture, learning, parenting There's a path to get people (children included) to get into technology with enough of the veneer of convenience to make sure it is a learning experience... While keeping it pleasurable. <https://havenweb.org/2026/05/28/retro-tech.html> Pokémon Go Scans Quietly Trained The Navigation Tech Now Headed Into Military Drones Tags: tech, game, surveillance, attention-economy, defense How do you like our particular brand of dystopia? That's what you get for using proprietary data farming game I guess. <https://dronexl.co/2026/06/09/pokemon-go-scans-niantic-vantor-military-drone-navigation/> The Blight Reaches Microsoft: 73 Repos Disabled in 105 Seconds Tags: tech, microsoft, github, security, supply-chain, ai, machine-learning, gpt, copilot There's really something nasty at play. Those coding agents are clearly not insulated from the system enough and too easy to manipulate in order to exfiltrate sensitive information. <https://opensourcemalware.com/blog/miasma-reaches-azure> our workplace LLM mass delusion Tags: tech, ai, machine-learning, gpt, management, trust This piece asks a very profound question in fact. If you're in a workplace where senior management allows and pushes everyone to get deluded about the real capabilities of those tools, how do you later move forward and rebuild trust? <https://blog.avas.space/llm-circus/> To my students Tags: tech, learning, culture, ethics, politics, quality Very nice piece, timely and needed. Indeed, let's hope people stick to those principles. <http://ozark.hendrix.edu/~yorgey/forest/00FD/index.xml> How LLMs Actually Work Tags: tech, ai, machine-learning, gpt, architecture, neural-networks A good primer on the main architecture traits of transformer models. <https://www.0xkato.xyz/how-llms-actually-work/> Local-First Software Is Easier to Scale Tags: tech, performance, architecture It's definitely easier not having to scale at all. Which is what you get when you design for local first / client side. <https://elijahpotter.dev/articles/local-first-software-is-easier-to-scale> Linux latency measurements and compositor tuning Tags: tech, graphics, linux, desktop, performance, debugging Interesting read, this is really tricky to measure such latency. It looks like we might have room for improvements on latency still. Curious to see if the proposed fixes will make it in kwin. <https://farnoy.dev/posts/linux-latency> Test-case Reducers Are Underappreciated Debugging Tools Tags: tech, tests, debugging Interesting family of testing and debugging tools indeed. I should definitely reach out to those more. https://tratt.net/laurie/blog/2026/test_case_reducers_are_underappreciated_debugging_tools.html Why Queues Don't Fix Overload (And What To Do Instead) Tags: tech, queuing, architecture, distributed Queues are not magic. If they're unbounded you're in for a world of pain as load increases. <https://pmbanugo.me/blog/why-queues-dont-fix-overload-and-what-to-do-instead> The User Doesn't Care - But you should Tags: tech, programming, quality Indeed, when people say "users don't care about quality" (tests or otherwise), this is mostly folklore. As soon as something goes wrong they'll care. <https://lewiscampbell.tech/blog/260607.html> The un-hateable engineering managers Tags: tech, engineering, management Sometimes, you got to deliver the bad news... It's healthy if you feel uneasy about it though. <https://newsletter.manager.dev/p/the-un-hateable-engineering-managers> Bye for now!

- [Code and logic for tournaments in the Mankala Engine](#) (2026/06/11 17:02)

Structure of code and logic for tournaments in the Mankala Engine. So, we want the tournaments to be able to: Create custom game rooms (with different variants, number of players, and time limit) Join the rooms with room codes. Create elimination brackets for players. Have empty slots filled with computerized opponents. A scoreboard after the game for final rankings. The tournament is going to be part of the Multiplayer variant. For that, when a person clicks on Multiplayer game mode, they will be asked to select from the given two options of a casual 1 to 1 game or another a tournament. Then they will be asked to either create a room or join the room with the code. Key elements to implement: Room codes will be created by mapping XMPP MUC JIDs (e.g., A3F7K2 → tournament-a3f7k2@conference.server.com). Fill computerized opponents: When it is

detected that there is an absence of required players, the game will be filled with computer opponents in all the empty spots, just like we have for the vs AI game mode. Here is an example of how the tournaments.cpp can be implemented: `class TournamentManager : public QObject { Q_OBJECT Q_PROPERTY(QVariantList tournaments READ getTournaments NOTIFY tournamentsChanged) Q_PROPERTY(QVariantMap activeTournament READ getActiveTournament NOTIFY activeTournamentChanged) public: // --- Tournament duration --- Q_INVOKABLE QString createTournament(const QString& variant, // "Bohnenpiel", "Oware", "Pallanguli" int maxPlayers, // eg. 4, 8, 16 int durationDays, // tournament duration const QString& organizerJid // creator's XMPP JID); Q_INVOKABLE void joinTournament(const QString& roomCode, const QString& playerJid); Q_INVOKABLE void startTournament(const QString& tournamentId); // --- Match management --- Q_INVOKABLE void reportMatchResult(const QString& matchId, const QString& winnerJid); Q_INVOKABLE void advanceBracket(const QString& tournamentId); // --- Computer Opponents --- Q_INVOKABLE void fillEmptySlots(const QString& tournamentId); // --- Room codes --- Q_INVOKABLE QString generateRoomCode(); Q_INVOKABLE QString resolveRoomCode(const QString& code); signals: void tournamentsChanged(); void activeTournamentChanged(); void matchReady(const QString& matchId, const QString& player1, const QString& player2); void tournamentComplete(const QString& tournamentId, const QVariantList& rankings); void playerJoined(const QString& playerJid); };` For filling empty spaces we can use the logic as: `struct Match { QString matchId; QString player1Jid; // empty string = unfilled slot QString player2Jid; QString winnerJid; int round; // 0 = first round, 1 = quarter-finals, etc. int position; // position within the round bool isAIMatch; // true if one/both players are computer };` class TournamentBracket { public: void initialize(int playerCount); // builds bracket tree void seedPlayers(const QStringList& playerJids); void fillWithAI(); // fills empty slots Match getNextMatch() const; void recordResult(const QString& matchId, const QString& winnerJid); bool isComplete() const; QVariantList getRankings() const; private: std::vector<std::vector<Match>> m_rounds; // rounds[0] = first round, etc. int m_totalRounds; }; At the end of the elimination cycle and the tournament bracket, we can get the results and show them on our created leaderboard page with the XMPP icons, usernames, and the number of games the players won. What more can be added? I plan to research more on whether we can store and display these games in the player profiles and create a communication channel for the players and spectators during the ongoing game, which could be both text and voice chat using XMPP and other relevant communication protocols. Thanks for reading. □

- [Introducing Qt's Figma Design System Extraction Skills for Developers](#) (2026/06/11 04:59)

Recreating a design system manually in QML is a laborious task for a Qt developer. A typical Figma design system can include hundreds of design tokens for colors, typography, spacing, radii, shadows, and motion durations - plus dozens of UI components, each with multiple variants and states. Every value has to be transcribed precisely, and even small mismatches can quietly desynchronize the implementation from the Figma source. Two new AI skills close the design-system-to-code gap between Figma and Qt. The Qt Figma Token Extraction skill converts your Figma design tokens directly into QML singletons. The Qt Figma Component Generation skill then turns your Figma component library into idiomatic QML controls that consume those singletons. Together they automate the full design-system handoff. The skills delegate this entire workflow to an AI agent, which connects to Figma, reads the design system, and produces clean, idiomatic QML ready to drop into a Qt project.

- [KDE Ships Frameworks 6.27.0](#) (2026/06/11 00:00)

Thursday, 11 June 2026 KDE today announces the release of KDE Frameworks 6.27.0. This release is part of a series of planned monthly releases making improvements available to developers in a quick and predictable manner. New in this version Baloo [Query] Make Query::exec() idempotent. Commit. [autotests] Remove unnecessary/unused include. Commit. Tighten used namespaces in implementations. Commit. Term:

Remove unused userData member and methods. Commit. Bluez Qt Enable LSAN in CI. Commit. Add Qt6DBus to public link interface. Commit. Breeze Icons Configure_package_config_file: drop unused PATH_VARS CMAKE_INSTALL_PREFIX. Commit. Replace Acrobat branding with Okular iconography. Commit. Fixes bug #487595 Remove im-[company logo] icons. Commit. See bug #487595 Remove acroread icon. Commit. See bug #487595 Extra CMake Modules Set INTERFACE_LINK_LIBRARIES if the found library is a static library. Commit. Fix: translation loading location issue on macOS. Commit. ECMDeprecationSettings: Set QT_DEPRECATED_WARNINGS_SINCE to current version when newer warnings should not be shown. Commit. Try harder to determine a good default for the Android API level. Commit. Silence more warnings in python bindings. Commit. Add note to ECMGenerateQmlTypes that it's basically obsolete. Commit. Add KDE_INSTALL_QTMETATYPESDIR. Commit. Sanitizer: add support for realtime sanitizer. Commit. KArchive Allow to set CMake modules path outside additionally. Commit. Use default DEFAULT_SEVERITY for logging. Commit. KAuth KF6AuthCore: move Qt6::Gui to private link interface. Commit. Port to KWaylandExtras::xdgActivationToken. Commit. KBookmarks KF6BookmarksWidgets: list KF6::Bookmarks in public interface, not private. Commit. KCMUtils Add support for Union to SettingHighlighterPrivate. Commit. Kcmshell: Don't leak QML engine. Commit. Add smoke-test option to kcmshell. Commit. KCodecs KEmailAddress::decodeMailtoUrl: return empty string on non-mailto URL. Commit. Fixes bug #519721 Remove unused code. Commit. KCharsets::fromEntity: bound numeric reference to QChar range and reorder bounds check. Commit. Fixes bug #519719. Fixes bug #519720 [Codec] Change compare method in charset lookup. Commit. [Codec] Fix infinite loop in charset lookup. Commit. Fixes bug #520200 KConfig Do not inherit publicly from std::map. Commit. KConfigSkeleton: Fix reading value for deleted entry with system default. Commit. Fixes bug #509416 KConfigXT: Add addItemPathList and addItemUrlList. Commit. CMake config file: check private QtDBus dependency only for static builds. Commit. Drop unneeded moc include. Commit. Fix build warnings with -Wnoexcept. Commit. Fixes bug #384583 Add autotest for KCoreConfigSkeleton::useDefaults. Commit. Kconfig_compiler: Fix QML forward declarations when using a namespace. Commit. Fixes bug #510091 Kwindowstatesaverquick: Do not force-show windows. Commit. CMake config file: remove no longer used Qt6Xml dep with (static) libs. Commit. Add test for ADMIN_ACCOUNT environment variable override. Commit. Fix wrong usage of qEnvironmentVariable function. Commit. CMake config file: add missing Qt6Core & Qt6Gui dependency check. Commit. Kconfig_compiler: Consistently accept lowercase type names. Commit. Install Qt metatypes. Commit. KConfigWidgets Hide windowsvista style from KStlyeManager. Commit. See bug #507942 Also activate window in KConfigDialog::showDialog. Commit. Add Amharic kf6_entry.desktop. Commit. KContacts CMake config file: add missing Qt6Gui dependency check. Commit. Impp: Try to always return a label for an IM protocol, even if none is known. Commit. Impp: Add helper getter to get only username portion of the address. Commit. Fix build not including new files in resource. Commit. Improtocols: Add a bunch of modern services. Commit. KCoreAddons Fix Clang-Tidy: Static member accessed through instance. Commit. Fix Clang-Tidy: Method 'initTestCase' can be made static. Commit. Fix Clazy: Unused QString lineFeed. Commit. KAboutData: Reload applicationData when desktopFileName param is set. Commit. See bug #459986 Fix build warnings with -Wnoexcept. Commit. Kaboutdata: Use application display name for --author. Commit. Klistopenfilesjob_unix: Remove unneeded qOverload for connect. Commit. Licenses Free Software Foundation has changed a bit the text of the licenses. Commit. KTextToHTML: escape HTML entities in highlighted text. Commit. Fixes bug #504078 Use default DEFAULT_SEVERITY for logging. Commit. Force locale on Windows as well for the KAboutData unit test. Commit. Install Qt metatypes. Commit. KCrash Add Android Sentry integration. Commit. KDav Use Depth: 1 instead of Depth: 0 for multiget REPORT requests. Commit. Fix percent-encoding of item hrefs in CalDAV/CardDAV multiget requests. Commit. Add job to create DAV collection. Commit. Autotests: Add davitemsfetchjobtest. Commit. Test: Remove manual test server. Commit. Enums: Deprecate GroupDav protocol support. Commit. KDeclarative Add location to

CalendarEvents::EventData. Commit. Add strict qmlint configuration. Commit. Use id instead of parent. Commit. Use Application instead of Qt.application. Commit. Remove unused include. Commit. Register QKeySequence to QML. Commit. Add missing module dependency. Commit. Access local property directly. Commit. Fix unqualified property access. Commit. Use int instead of enum as property type. Commit. Port to KI18nContext QML API. Commit. KDE Daemon Use default DEFAULT_SEVERITY for logging. Commit. KDE SU Make sure to handle \\0 without overflowing read. Commit. KGlobalAccel Add KGlobalAccel::setInverseShortcutActions() to the API. Commit. Format with clang-format and xmlint. Commit. Add empty lines in dbus interface files. Commit. Properly mark all deprecated APIs in org.kde.KGlobalAccel interface. Commit. Mark org.kde.KGlobalAccel.actionList as deprecated. Commit. Make KGlobalAccel::stealShortcutSystemwide() steal all matching shortcuts. Commit. Support a custom bus. Commit. KGuiAddons Install Qt metatypes. Commit. KHolidays Add Bolivian Holidays. Commit. Fixes bug #520179 .clang-tidy - fix typo StringComparisonFunctions. Commit. Src/lunarphase.cpp - crazy excludeall typedefs. Commit. CMakeLists.txt - update the flex and bison urls. Commit. Src/sunevents_p.h - fix include guard name. Commit. SunriseSet: place deprecation attribute standard-type-clang-compatible. Commit. Fix holidays appear twice in january. Commit. Fixes bug #498462 UA: Update Ukrainian holidays. Commit. .gitlab-ci.yml - don't fail on clang-tidy issues. Commit. KI18n Avoid use of Qt module header includes with QtQmlIntegration. Commit. Add "gnuintl" as a variant for Intl library name. Commit. Convert old prg doc reference to proper link. Commit. KIconThemes KIconButton: add some unit test for the properties. Commit. Widgets/kicondialog: clear custom when pressing OK. Commit. Fixes bug #520007 KImageformats Updated documentation on memory usage. Commit. JXR: add a memory allocation barrier in jxrlib. Commit. EXR: reject files with dimensions exceeding 300kx300k pixels. Commit. OSS Fuzz: set image allocation limit to 2000 MiB. Commit. Improve size limits for AVIF, HEIF, and RAW plugins. Commit. IFF: fix byte swapping in 16-bit DEEP images. Commit. Add Farbfeld read only support. Commit. DDS: fix mime type. Commit. IFF: DEEP image support. Commit. ImageAlloc: add image initialization support. Commit. Fix uninitialized value. Commit. HEIF: image transformation support. Commit. Update CI image. Commit. KIO KUrlNavigator: Use QStyle for navbar focus indication instead of custom painting. Commit. Knewfilemenu: refactor mimeType and icon. Commit. Knewfilemenu: delete EntryInfo and merge it into Entry. Commit. Knewfilemenu: add QDebug overload for KNewFileMenuSingleton::Entry. Commit. Knewfilemenu: reorder variables and prune redundancies. Commit. KDirModel: fix assert when typing file.png/a by mistake. Commit. Kmountpoint: Iterate mountpoint backwards to handle systemd-automount. Commit. Fixes bug #518012 Kioworker/file: avoid huge stack buf in file copy read/write fallback. Commit. Openurljob: respect "open in application" preference for binaries. Commit. Fixes bug #519773 KUrlNavigator: Fix double background interference when in editable mode. Commit. Fixes bug #506177. Fixes bug #520249. See bug #518285 KDirModel: Use mostLocalUrl for old url. Commit. Fixes bug #509461 KIO::CacheControl enum: place deprecation attribute standard-type-compatible. Commit. Kprocessrunner: For apps running in terminal, use terminal emulator's appld. Commit. Fixes bug #459986 Fix build warnings with -Wnoexcept. Commit. KFilePlaceEditDialog: deprecate for external usage. Commit. API docs: list kservice as dep, so KService types get links. Commit. KIO::OpenWith: work around need for include of KService in public header. Commit. KUrlNavigatorPlacesSelector: set accessible name. Commit. Kurlauthorized.h: drop unneeded include. Commit. Trash: Don't assert in parseURL. Commit. Davjob: Make it possible to use QXmlStreamWriter output. Commit. Davjob: Add MKCOL job. Commit. Dropjob: Sanitize input URLs. Commit. Fixes bug #518481 Kfileitem: Assert that URL scheme is not empty. Commit. See bug #518481 DropJobTest: Create cache dir when needed. Commit. KDirModel: Remove setting node preview on itemChanged. Commit. Fixes bug #517607 Don't show an error when the user cancels the device mounting password dialog. Commit. Fixes bug #400420 Knewfilemenu: centralize new file parsing. Commit. Don't try showing error message if it's empty. Commit. Configure_package_config_file: drop unused PATH_VARS KDE_INSTALL_DBUSINTERFACEDIR. Commit. Http

worker: call redirection also for redirectToTrailingSlash. Commit. Fixes bug #517682 Kurlnavigator: Don't show context menu targeting hidden buttons when path is compressed. Commit. Fixes bug #505197 Global: map KJob::NoError to ERR_NONE. Commit. Use default DEFAULT_SEVERITY for logging. Commit. Upgrade logging from info to warning where appropriate. Commit. Widgets: adapt dimensions of KOpenWithDialog on rotated monitors. Commit. Kdiroperator, kfilewidget: Use zoomsteps when using scrollwheel. Commit. Fixes bug #494704 Searchproviders: improve osm. Commit. Searchproviders: added osm. Commit. Searchproviders: added keyword app to flathub. Commit. Searchproviders: added "Flathub". Commit. Searchproviders: add "wikipedia" keyword. Commit. Searchproviders: localize query for german wikipedia. Commit. Searchproviders: added "ZDF Mediathek". Commit. Searchproviders: added "ARD Mediathek". Commit. Core/copyjob: Prompt to replace MS-DOS reserved chars for MTP. Commit. Fixes bug #489288 Kirigami Remove polyfill imports for SafeArea API. Commit. Don't apply safe area margin to page bottom when there's a footer. Commit. Only apply safe area margins when we have a parent. Commit. Fix top margin in Page overlayItem. Commit. Heading: set the color to Platform.Theme.textColor for heading. Commit. Explicitly layout the window content item vertically as well. Commit. Fix toolbar size with non-zero SafeArea margins. Commit. Controls/Badge: avoid binding loop. Commit. Fix access to ScenePosition attached property in DrawerHandle. Commit. GlobalDrawer: top-align topContent. Commit. Fixes bug #520337 Add basic tests for some components. Commit. Don't handle back key twice. Commit. FormLayout: Fix FormData namespace alias. Commit. Fixes bug #520076 Revert "Disable qmlint for some modules". Commit. Form: Guard root.Forms.FormAlignmentGroup.group. Commit. GlobalDrawer: Use implicitWidth, set preferredHeight in item. Commit. Fixes bug #517091 Remove unused import. Commit. Clean up internal imports. Commit. Controls: Remove defunct onCompleted block in AbstractApplicationWindow. Commit. Controls: Don't break width/height bindings in AbstractApplicationWindow. Commit. Remove duplicate ecm_qt_install_logging_categories. Commit. Use default DEFAULT_SEVERITY for logging. Commit. Action: Notify fromQActinChanged when the QAction changed. Commit. Form: Introduce FormAlignmentGroup. Commit. Install Qt metatypes. Commit. ToolbarLayout: don't crash when moreButton is not valid. Commit. Fixes subtitle padding calculation in forms flat mode. Commit. KMime New in this release KNewStuff EngineBase::addProvider(provider): fix consumer-side hiding virtual method. Commit. KNotifications Android: Improve icon handling. Commit. Android: Confirm reception of inline replies. Commit. Android: Set Intent package names rather than invalid component names. Commit. Android: Fix invalid iterator access. Commit. Don't update notifications before they have actually been sent. Commit. Android: Don't defer NotifyByAndroid::notify execution. Commit. Configure_package_config_file: drop unused PATH_VARS KDE_INSTALL_DBUSINTERFACEDIR. Commit. KQuickCharts Use default DEFAULT_SEVERITY for logging. Commit. KRunner Extract and install Qt metatypes. Commit. RunnerManager: move spurious member into pimpl class. Commit. ResultsModel: Directly call toInt from QVariant to avoid implicit conversion. Commit. RunnerManager: Avoid weird use of mutable iterator in favor of simple for loop. Commit. RunnerContext: Use QHash for unique IDs since order is not relevant. Commit. Use std::min/max instead of QT macros. Commit. KSVG Svg: clear stylesheetOverride on imageSetChanged. Commit. Fixes bug #503671 Svg: make eventFilter public again. Commit. CMake config file: remove duplicated targets include. Commit. Drop accidental duplicate QML registration. Commit. Remove unused ECM module. Commit. Add basic test for SvgItem. Commit. KTextEditor Use expanded document mode tabs in config pages. Commit. Revert "CI: Comment out Qt 6.12 for now since tests are failing". Commit. ScriptTester: Due to the change in Qt 6.12, getMessage() should return an empty string instead of undefined when there is no message. Commit. Add public API for visible/real line number conversion. Commit. Calculate which line numbers need repainting correctly. Commit. Allow disabling "use first line of doc as file name" for untitled docs. Commit. Fixes bug #519242 Use default DEFAULT_SEVERITY for logging. Commit. KTextWidgets KPluralHandlingSpinBox: place deprecation attribute standard type compatible. Commit. KUnitConversion Mark

US pints as common, rather than Imperial ones. Commit. KUserFeedback Configure_package_config_file: drop unused PATH_VARS KDE_INSTALL_INCLUDEDIR. Commit. KWidgetsAddons Use version-controllable deprecation warning macros instead of hardcoding. Commit. Update kchaselect-data to Unicode 17.0. Commit. Force locale on Windows as well for unit test. Commit. Use default DEFAULT_SEVERITY for logging. Commit. KWindowSystem KKeyServer: place deprecation attribute standard-type-clang-compatible. Commit. Use default DEFAULT_SEVERITY for logging. Commit. Kwaylandextras: refine activation token documentation a bit. Commit. KXMLGUI Kxmlguiwindow: Special case KToolBarPopupAction for CommandBar. Commit. Kmainwindow commandbar: include root ActionGroup actions. Commit. Avoid setting session management window properties without session management. Commit. Fixes bug #519786 Add file suffix for shortcuts. Commit. Modem Manager Qt Use default DEFAULT_SEVERITY for logging. Commit. Network Manager Qt Fix bridgeport hairpin-mode setting key. Commit. Use default DEFAULT_SEVERITY for logging. Commit. Oxygen Icons New in this release Purpose CMake config file: add missing Qt6Gui & Qt6Widgets dependency check. Commit. Change Pastebin to Pastebin.com. Commit. Adapt to KOnlineAccounts interface change. Commit. Add missing find_dependency for Qt6QmlIntegration. Commit. Use default DEFAULT_SEVERITY for logging. Commit. Adapt to KOnlineAccounts API change. Commit. Replace var with const. Commit. Remove unused import. Commit. Convert org.kde.purpose.private.widgets to proper QML module. Commit. Fix unqualified property lookup. Commit. Rename JobView::state to jobState. Commit. Use KI18nContext. Commit. Use fully qualified typename in property. Commit. Fix deprecation warnings from ECM. Commit. Remove CMake warning about arc runtime dependency. Commit. QQC2 Desktop Style README.md kdesrc-build -> kde-builder. Commit. Remove group separators from SpinBox by default like QSpinBox. Commit. Solid Use globally-configured storage size units for labeling disks. Commit. Fixes bug #518493 Allow to set CMake modules path outside additionally. Commit. Kdeconnect: Always remove device before emitting deviceRemoved. Commit. Use default DEFAULT_SEVERITY for logging. Commit. Sonnet CMake config file: fix missing Qt6Widgets check with SONNET_USE_WIDGETS:ON. Commit. Allow to set CMake modules path outside additionally. Commit. Remove unused ECM module. Commit. Syndication CMake config file: add missing Qt6Core dependency check. Commit. Syntax Highlighting Add missing license texts. Commit. YAML: fix quoted no-string in list. Commit. Fixes bug #500809 Add syntax highlighting for Slint files. Commit. Minor fixes on gemtext syntax and reference. Commit. Add BNF and ABNF syntax highlighting. Commit. Fixes bug #518109 Breeze Dark theme: less vibrant color for BracketMatching to make dsFunction color visible. Commit. Fixes bug #519805 Theme_contrast_checker.py: use Okl as default color space. Commit. Theme_contrast_checker.py: add --neighbor-color and --neighbor-distance to filter colors that are close to a specific color. Commit. Theme_contrast_checker.py: add --bg-color to view colors of a theme with a specific background color. Commit. Fish: update to latest version and fix various bugs. Commit. Fixes bug #519060 LaTeX: fix optional parameter with minted shortcuts. Commit. Fixes bug #514536 J: update to latest version from github. Commit. Fixes bug #518108 Raku: bump highlighting definition version. Commit. Raku: Allow - inside \c[], fill out allowed backslashes inside char classes. Commit. Raku: Fix breakage of double backslash before ending single quote. Commit. Raku: Allow signatures on token/rule/regex declarations. Commit. Threadweaver Remove duplicate headers. Commit.

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

Permanent link:
<https://wiki.tromjaro.alexio.tf/doku.php?id=news:planet:kde&rev=1583653480>

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

