KDE Planet - Latest News

- This Week in Plasma: UI and performance improvements (2025/11/22 00:01)
  Welcome to a new issue of This Week in Plasma! This week there were many user interface and performance improvements — some quite consequential. So let’s get right into it! Notable New Features Plasma 6.6.0 Windows can now be selectively excluded from screen recording! This can be invoked from the titlebar context menu, Task Manager context menu, and window rules. (Stanislav Aleksandrov, link) Notable UI Improvements Plasma 6.6.0 With a dark color scheme, the blur effect now produces a blur that’s darker (ideally back to the level seen in Plasma 6.4) and also more vibrant in cases where there are bright colors behind it. People seemed to like this! But for those who don’t, the saturation value of the blur effect is now user-configurable, so you can dial it in to your preferred level. (Vlad Zahorodnii, link 1, link 2, and link 3) When clicking on grouped Task Manager icons to cycle through their windows, full-screen windows will no longer always be raised first. Now, windows will be raised in the order of their last use. (Grégori Mignerot, link) Did a round of UI polishing on the portal remote control dialog to make it look better and read more naturally. (Nate Graham and Joshua Goins, link 1 link 2, link 3 and link 4) When you open the Kickoff Application Launcher and your pointer happens to end up right on top of one of the items in the Favorites view, it won’t be selected automatically. (Christoph Wolk, link) The Kickoff Application Launcher widget now tries very hard to keep the first item of the search results view selected — at least until the point where you focus the list and start navigating to another item. (Christoph Wolk, link) Discover now uses more user-friendly language when it’s being used to find apps that can open a certain file type. (Taras Oleksy, link) You’re now far less likely to accidentally raise an unintended app when a notification happens to appear right underneath something you’re dragging-and-dropping. (Kai Uwe Broulik, link) KMenuEdit now lets you select multiple items at a time for faster deletion. (Alexander Wilms, link) The QR code dialog invokable from the clipboard has been removed, and instead the QR code is shown inline in the widget. This makes it large enough to actually use and also reduces unnecessary code. (Fushan Wen, link) Notable Bug Fixes Plasma 6.5.3 Fixed a rare case where KWin could crash when the system wakes from sleep. (Xaver Hugl, link) Worked around a QML compiler bug in Qt that made the power and session buttons in the Application Launcher widget overlap with the tab bar if you resized its popup. (Christoph Wolk, link) Plasma 6.5.4 Fixed a regression in menu sizing that got accidentally backported to Plasma 6.5.3. All should be well in 6.5.4, and some distros have backported the fix already. (Akseli Lahtinen and Nate Graham, link) Fixed a Plasma 6 regression that broke the ability to activate the System Tray’s expanded items popup with a keyboard shortcut. (Cursor AI, operated by Mikhail Sidorenko, link) Fixed a regression caused by a Qt change that broke the clipboard’s Actions menu from being able to appear when the configuration dialog wasn’t open. (Fushan Wen, link) Fixed a bug that could make the Plasma panel’s custom size chooser appear on the wrong screen. (Vlad Zahorodnii, link) Fixed a bug that could make the clipboard contents get sent many times when it’s being set programmatically in a portal-using app. (David Redondo, link) Fixed a memory leak in Plasma’s desktop. (Vlad Zahorodnii, link) Fixed a memory leak in the clipboard Actions menu. (Fushan Wen, link) KWin’s zoom effect now saves its current zoom level a little bit after you change it, rather than at logout. This prevents a situation where the system is inappropriately zoomed in (or not zoomed in) after a KWin crash or power loss. (Ritchie Frodomar, link) Fixed a bug that made the optional Textual List representation of multiple windows in the Task Manager widget fail to get focus when using medium focus stealing prevention. (David Redondo, link) Plasma 6.6.0 Worked around a bug in some XWayland-using games that made it impossible type text

into certain popups. (Xaver Hugl, link) Clearing KRunner's search history now takes effect immediately, rather than only after KRunner was restarted. (Nate Graham, link) With a very narrow display and a high scale factor, the buttons on the login, lock, and logout screens can no longer get cut off; now they wrap onto the next line. (Nate Graham, link) Frameworks 6.21 Fixed a bug that could confuse KWallet — when being used as a Secret Service proxy for KeePassXC — into becoming convinced that it needed to create a new wallet. (Marco Martin, link) Fixed two memory leaks affecting QML-based System Settings pages. (Vlad Zahorodnii, link 1 and link 2) Other bug information of note: 4 very high priority Plasma bugs (Same as last week). Current list of bugs 34 15-minute Plasma bugs (Up from 31 last week). Current list of bugs Notable in Performance & Technical Plasma 6.5.3 Apps that use the Keyboard Shortcuts Portal to set shortcuts can now remove them in the same way. (David Redondo, link) You can now use Spectacle's Active Window mode to take a screenshot of WINE windows. (Xaver Hugl, link) Plasma 6.6.0 Made a major improvement to the smoothness of animations throughout Plasma and KWin for people using screens with a refresh rate higher than 60 Hz! (David Edmundson, link) Reduced the amount of unnecessary work KWin does during its compositing pipeline. (Xaver Hugl, link) When you delete a whole category's worth of shortcuts on System Settings' Shortcuts page, all the shortcuts get grayed out and cease to be interactive, and a warning message tells you they'll seen be deleted and gives you a chance to undo that before it happens. (Nate Graham, link) Frameworks 6.21 KConfig now parses config files in a stream rather than opening them all at once, which allows it to notice early when a file is corrupted or improperly formatted. This prevents freezes in several places. (Méven Car, link 1, link 2, and link 3) When using the Systemd integration functionality (which is on by default if Systemd is present), programs will no longer fail to launch while there are any environment variables beginning with a digit, as this is something Systemd doesn't support. (Christoph Cullmann, link) How You Can Help Donate to KDE's 2025 fundraiser! It really makes a big difference. Believe it or not, we've already hit out our €75k stretch goal and are €5k towards the final one. I'm just in awe of the generosity of the KDE community and userbase. Thank you all for helping KDE to grow and prosper! If money is tight, you can help KDE by directly getting involved. 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. To get a new Plasma feature or a bugfix mentioned here, feel free to push a commit to the relevant merge request on invent.kde.org.

- Web Review, Week 2025-47 (2025/11/21 10:46)
  Let's go for my web review for the week 2025-47. In 1982, a physics joke gone wrong sparked the invention of the emoticon - Ars Technica Tags: tech, history, culture If you're wondering where emoticons and emojis are coming from, this is a nice little piece about that. https://arstechnica.com/gadgets/2025/11/in-1982-a-physics-joke-gone-wrong-sparked-the-invention-of-the-emoticon/ Screw it, I'm installing Linux Tags: tech, linux, foss, gaming Clearly something is brewing right now. We're seeing more and more people successfully switching. https://www.theverge.com/tech/823337/switching-linux-gaming-desktop-cachyos Lawmakers Want to Ban VPNs—And They Have No Idea What They're Doing Tags: tech, vpn, privacy, law This is totally misguided… Let's hope no one will succeed passing such dangerously stupid bills. https://www.eff.org/deeplinks/2025/11/lawmakers-want-ban-vpns-and-they-have-no-idea-what-theyre-doing Learning with AI falls short compared to old-fashioned web search Tags: tech, ai, machine-learning, gpt, learning, teaching If there's one area where people should stay clear from LLMs, it's definitely when they want to learn a topic. That's one more study showing the knowledge you retain from LLMs briefs is shallower. The friction and the struggle to get to the information is a feature, our brain needs it to remember properly. https://theconversation.com/learning-with-ai-falls-short-compared-to-old-fashioned-web-search-269760 The Psychogenic Machine: Simulating AI Psychosis, Delusion Reinforcement and Harm Enablement in Large Language Models Tags: tech, ai, machine-learning, gpt, psychology, safety

The findings in this paper are chilling… especially considering what fragile people are doing with those chat bots.
https://arxiv.org/abs/2509.10970v1 Feeds, Feelings, and Focus: A Systematic Review and Meta-Analysis Examining the Cognitive and Mental Health Correlates of Short-Form Video Use Tags: tech, social-media, cognition, psychology Unsurprisingly the news ain't good on the front of social media and short form videos. Better stay clear of those. https://psycnet.apa.org/fulltext/2026-89350-001.html Do Not Put Your Site Behind Cloudflare if You Don't Need To Tags: tech, cloud, decentralized, web Friendly reminder following the Cloudflare downtime earlier this week.
https://huijzer.xyz/posts/123/do-not-put-your-site-behind-cloudflare-if-you-dont Cloudflare outage on November 18, 2025 Tags: tech, cloud, complexity, safety, rust Wondering what happened at Cloudflare? Here is their postmortem, this is an interesting read. Now for Rust developers… this is a good illustration of why you should stay clear from unwrap() in production code. https://blog.cloudflare.com/18-november-2025-outage/ Needy Programs Tags: tech, ux, notifications Kind of ignore the security impact of the needed upgrades, but apart from this I largely agree. Most applications try to push more features in your face nowadays, unneeded notifications and all… this is frankly exhausting the users.
https://tonsky.me/blog/needy-programs/ I think nobody wants AI in Firefox, Mozilla Tags: tech, browser, ai, machine-learning, gpt, mozilla Looks like Mozilla is doing everything it can to alienate the current Firefox user base and to push forward its forks.
https://manualdousuario.net/en/mozilla-firefox-window-ai/ DeepMind's latest: An AI for handling mathematical proofs Tags: tech, ai, machine-learning, mathematics, google That's an interesting approach. Early days on this one, it clearly requires further work but it seems like the proper path for math related problems. https://arstechnica.com/ai/2025/11/deepminds-latest-an-ai-for-handling-mathematical-proofs/ Production-Grade Container Deployment with Podman Quadlets Tags: tech, systemd, containers, linux, system, podman Podman is really a nice option for deploying containers nowadays. https://blog.hofstede.it/production-grade-container-deployment-with-podman-quadlets/ Match it again Sam Tags: tech, regex, rust Nice alternative syntax to the good old regular expressions. Gives nice structure to it all. There's a Rust crate to try it out.
https://www.sminez.dev/match-it-again-sam/ 10 Smart Performance Hacks For Faster Python Code Tags: tech, python, performance Some of this might sound obvious I guess. Still there are interesting lesser known nuggets proposed here.
https://blog.jetbrains.com/pycharm/2025/11/10-smart-performance-hacks-for-faster-python-code/ Floodfill algorithm in Python Tags: tech, python, algorithm, graphics This is a nice little algorithm and it shows how to approach it in Python while keeping it efficient in term of operations.
https://mathspp.com/blog/floodfill-algorithm-in-python AMD vs. Intel: a Unicode benchmark Tags: tech, amd, intel, hardware, simd, performance Clearly AMD is now well above Intel in performance around AVX-512. This is somewhat unexpected.
https://lemire.me/blog/2025/11/16/amd-vs-intel-a-unicode-benchmark/ Memory is slow, Disk is fast Tags: tech, memory, storage, performance, system No, don't go assuming you can use disks instead of ram. This is not what it is about. It shows ways to get more out of your disks though. It's not something you always need, but sometimes it can be a worth endeavor. https://www.bitflux.ai/blog/memory-is-slow-part2/ Compiler Options Hardening Guide for C and C++ Tags: tech, c++, security Good list of hardening options indeed. That's a lot to deal with of course, let's hope this spreads and some defaults are changed to make it easier.
https://best.openssf.org/Compiler-Hardening-Guides/Compiler-Options-Hardening-Guide-for-C-and-C++.html The problem with inferring from a function call operator is that there may be more than one Tags: tech, c++, type-systems, safety The type inference in C++ can indeed lead to this kind of traps. Need to be careful as usual. https://devblogs.microsoft.com/oldnewthing/20251002-00/?p=111647 There's always going to be a way to not code error handling Tags: tech, programming, safety, failure Depending on the ecosystem it's more or less easy indeed. Let's remember that error handling is one of the hard problems to solve.

https://utcc.utoronto.ca/~cks/space/blog/programming/AlwaysUncodedErrorHandling Disallow code usage with a custom clippy.toml Tags: tech, rust, tools, quality Didn't know about that clippy feature. This is neat, allows to precisely target some of your project rules. https://www.schneems.com/2025/11/19/find-accidental-code-usage-with-a-custom-clippytoml/ The Geometry Behind Normal Maps Tags: tech, 3d, graphics, shader Struggling to understand tangent space and normal maps? This post does a good job to explain where this comes from. https://www.shlom.dev/articles/geometry-behind-normal-maps/ Know why you don't like OOP Tags: tech, object-oriented I don't get why object oriented programming gets so much flack these days… It brings interesting tools and less interesting ones. Just pick and choose wisely like for any other paradigm. https://zylinski.se/posts/know-why-you-dont-like-oop/ Ditch your (mut)ex, you deserve better Tags: tech, multithreading, safety If you're dealing with multithreading you should not turn to mutexes by default indeed. Consider higher level primitives and patterns first. https://chrispenner.ca/posts/mutexes Brownouts reveal system boundaries Tags: tech, infrastructure, reliability, failure, resilience Interesting point of view. Indeed, you probably want things to not be available 100% of the time. This forces you to see how resilient things really are. https://jyn.dev/brownouts-reveal-system-boundaries/ Tech Leads in Scrum Tags: tech, agile, scrum, tech-lead, leadership Interesting move on the Scrum definitions to move from roles to accountabilities. The article does a good job explaining it but then falls back into talking about roles somehow. Regarding the tech leads indeed they can work in Scrum teams. Scrum don't talk about them simply because Scrum don't talk about technical skills. https://www.patkua.com/blog/tech-leads-in-scrum/ How to Avoid Solo Product Leadership Failure with a Product Value Team Tags: tech, agile, product-management I wonder what the whole series will give. Anyway I very much agree with this first post. Too often projects have a single product manager and that's a problem.
https://www.jrothman.com/mpd/2025/11/how-to-avoid-solo-product-leadership-failure-with-a-product-value-team-part-1/ Bye for now!

- FAQs (2025/11/21 00:00)
  Table of Contents Import Tool My camera is not in the list of supported devices How to use a USB Mass Storage Camera Run Time Issues Thumbnail Generation Fails on Large Files Why are my RAW images so dark? Some Properties in Sidebar are Unknown or Unavailable How can I inform you about bugs and wishes? What can I do if digiKam always crashes when doing something? How to deal with UTF-8 encoding issues? ExifTool not Found at Startup Database digiKam does not work when the album library is on a network share How to Import Tags from Another Program? How to Migrate a Database Between Different Computers? Customizations How to create a color theme? How can I change the default external video player? How can I change the text font size? How to change the single-click behavior on a thumbnail? How can I change the browser used with the More Info button from the map sidebar? Is digiKam available in my language? Distribution Package Pre-Compiled Bundles Workflow Which file format should I use? Contribute digiKam does not compile Skill Level: INTERMEDIATE
- Text Tool Phase 3 (2025/11/20 10:22)
  When I started work on the text tool, I had planned it in three phases. The first phase was going to be the on canvas editor. And I handled inserting and removing text, moving the cursor around, and creating simple wrapping areas. IME support and simple copy-paste was also handled during this phase. The second phase was about rich text editing. I created a text properties docker, implemented a font database, style presets, a glyph palette and went over each property individually to make 100% sure it worked. I wrote a bunch of blogposts about this in the past (text layout, fonts, OpenType, metrics, and more). Which brings us to phase three… Removing the old Rich Text editor and shortcuts. Previously, Krita used a dialog to allow rich text editing. The dialog happened because after several years of working on Calligra (which Krita was part of for a good while), none of the core devs were confident in their ability to conjure a full-featured on canvas rich text editor in a short

amount of time. Then, the plan became to have a SVG source editor as the main way to interact with text… But then we realized that SVG text is very verbose, even compared to the pretty verbose HTML. Furthermore, we only had a few weeks left, so we had to quickly put something together that could produce SVG output without being too alienating. The old rich text editor window, next to the preview on-canvas. In a dark mode, the theming issues become especially prevalent, here the text is black both in the editor as in the preview, but the background is dark grey. This was one of the core headaches with this editor. It's interesting to reflect back on this, we had assumed at the time that the artists using our program would be quite technically competent. Over the past few years however Krita has picked up so much steam that it frequently ends up being the program to teach people that there's such a thing as "working memory" and that one can run out of said memory when they, for example, try to make a 60 fps animation at 4K resolution. The rich text editor also didn't help matters here because the conversion to and from SVG text wasn't optimal: SVG 1.1 doesn't have a concept of lines or line wrapping, being more a graphical description format like PDF than say, a rich text document. Furthermore, there were endless issues with theming, font size handling, differences between Qt's rich text implementation and HTML overall, fonts, you name it. So, given that the first two phases left us with a functional on-canvas text editor that can style text, it felt good to remove the rich text editor. The source editor remains, because it can still do some advanced tricks if you know SVG very well, but for the vast majority of text tasks it is unnecessary. That left the shortcuts. Because the rich text editor was implemented with KXmlGui, each of the entries in the toolbars was a QAction that could be configured. I went and implemented the majority of those as shortcuts in the text tool. Because the majority of these shortcuts just changed a single property value, I took inspiration here from how the old artistic text tool was implemented, which is to say, using the setData() function on QAction. The setData takes a struct that contains simple instructions on which property to edit, and how to edit it (increase it, or set a specific value, for example), and this is then used to test the toggledness on the action or generate the appropriate property setting command. The shortcuts are a little interesting in that the text tool needs to do its own shortcut matching, instead of using the global system. This is because the text tool needs to be able to discern text input from shortcuts, as well, all QKeySequence::StandardShortcuts for text navigation and selection have been implemented directly into the text tool. The latter needed to be handled directly because if you are working with vertical text you will want to have the up/down keys be for navigating forward/backwards in the text. In practice, this means that the tool first tests non-modifier arrow keys and basic input. Then it checks the direction and writing mode of the text, and ensures that any directional keys get replaced by their expected variants for said direction and writing mode. It then checks the shortcuts and finally the standard sequences. This latter order is because some standard sequences use the same shortcuts that are typically reserved for property changes (for example, deleting a line and setting underline can both have the Ctrl + U shortcut). Only a few pre-existing shortcuts are supported right now, because I was feeling the end of the project looming. It is also a very good starting point for people who are interested in hacking on the text tool, as it makes you think about how the text properties function, so I have left it alone for now. Replacing the tool options. Next up was replacing the tool options. Tool options within Krita are specifically for changing the behaviour of the current tool, as well as accessing extra functionality, and Krita's text tool had one that allowed you to set some properties for new text creation, as well as accessing the separate editing dialog. The old tool options with a "create new text with" group of widgets and an "Edit Text" button. This was a QWidgets based UI element, and one of the things I've been doing with the text tool is that every UI element would be written in primarily QML, using QWidgets only as a fallback. Beyond that, we've been trying to use Lager to keep track of data-editing, and tool configuration is a good candidate for a lager model. The new options are both longer, because there's more toggles, but at the same time also simpler to use… As for the options themselves. Like the old options, they allow selecting the default text that new texts are being created with. However, instead of

replicating the text properties docker in its entirety, it now provides a drop down to select a style preset. Beyond that, there's a toggle to use the current presets in the text properties docker. The text properties docker itself also has a button to open it inside the tool options. This is because I observed multiple people trying to use the "new texts are created with" options (that were labelled as such) in the old widget to manipulate the selected text. Hopefully this'll guide people to the text properties docker. Then, there's the two options for the text tool itself: "Paste Rich Text By Default" and "Use Visual Cursor". The former is about whether Ctrl + V should paste the rich text or the plain text, while there's separate actions for pasting plain or rich text explicitly. For some reason the majority of word processors always paste rich text by default and never allow you to configure this, even though this is not how anyone wants to use a word processor. It therefore made sense to make this a toggle. The other is for bidirectional text. The default is the "logical" order of the bidirectional text, that is, the order in which you read it. This has always been a little bit of a headache to programmers, because you end up with a cursor that skips positions and will in some cases go into the opposite direction of the key you're pressing. This is why a lot of programs offer a "visual" cursor, which will try to follow the direction the keys are pressed in. This too is configurable, as it depends per person which of the two is more intuitive. Finally, there's several buttons, some for opening the dialogs like the Glyph Palette or the Source Editor, others for toggling Type Setting Mode and finally a set of buttons for the converter actions. More on these latter two in a bit. Annoyingly, QML cannot handle popups properly while used together with a QQuickWidget: the popup is clipped by the QQuickWidget bounds (specifically, it's internal QML scene). In Qt6 there's a toggle to not have Pop ups clipped to the scene, but that crashes in QApplication when used inside a QQuickWidget. This is a problem because it makes it hard to provide the style preset dropdown (or the font dropdown for that matter). I am trying to find a solution, but in the meantime I've made the delegates much smaller. The CSS style presets now have much more compact delegates to handle being inside a dropdown. I hope I'll be able to fix the pop-up problem in the coming weeks though, as this is kinda annoying. PSD text and vector support. This was a somewhat tangential project, and I had written most of it even before I had written the on canvas text editor. PSD vectors are stored as a vector mask on a fill. Because I had previously already written support for the fills, supporting vector masks meant supporting vector shapes. Then, figuring out which layer data describes the stroke, and which describes 'meta shapes' like rectangles, ellipses and stars was the rest. The vector masks, in particular PSD's path format, are interesting in that they have a particular floating point format. Luckily, Scribus already supported loading those paths, so I was able to reference their parsing code and concoct writing code based on it. Most of the work by far was making sure the transforms were 100% correct. I was helped here by Deif Lou, who provided me with a ton of test files to check against. Text was not as simple. Text in PSD is stored as a PDF structure, which, if you are unfamiliar with PDF, is format not dissimilar to JSON. You can see the way text is stored by opening up a PSD with text as a text file in a given simple text editor. Unlike most PSD data, which is stored in binary (and is best inspected with a hex-editor), the text data is largely stored as ascii text (with actual strings being stored in 16bit BE). Some inspection reveals that the text itself is stored in a range based manner. The whole plain text is written at the start, and then a list of character "sheets" is presented. After all those, a list of numbers of equal size, each presenting what range is occupied by the sheet with the same index. Idem for paragraph properties (PS, unlike SVG 2, can have multiple paragraphs. SVG 1.2 did have multiple paragraphs, but there's only one real implementation of SVG 1.2 (Inkscape) due its complexity). PSD text is also, notably, stored and sized in pixels, regardless whether the format is saying points (this is because within the Apple ecosystem the two units are the same, while outside it, a digital point is always 1/72th of an inch). But there were still a number of mysteries. Like, where were the text paths stored? And how about the advanced OpenType features? After I asked on Krita-Artists for sample files, I found the answer: There's another chunk of advanced text data at the end of the document. This second chunk is far more complex, and, worryingly, it uses numbers for the keys for more recent

versions of the Adobe products. Thankfully, it turns out that this second chunk is in fact shared between the Adobe suite, and the Inkscape project had already done a lot of reverse engineering. While I am now able to read this data and load text from it, there's still some missing information. It seems there's like, Line and Cluster-specific positioning and glyph info present in this extra data, and if you write it without said data, Photoshop will complain it is missing. I haven't had the time to look into this properly, so we can't write advanced text objects as of now. Another wrinkle is that text and vectors are per-layer in PSD, while in Krita each vector layer is its own SVG document. So I also had to write some code to explode the Krita vector layers so each shape is written as a separate PSD layer. Similarly, despite being able to load a lot, Krita's SVG+CSS based text layout is fundamentally different from Adobe's, so text loading isn't perfect. None the less, it should provide valuable for people's archive. Because of these differences, Krita will ask whether you want to load the text layers as text shapes or whether you want to load them as raster data. I do want to eventually return as figure out that final missing data. As well, because PSD doesn't do inheritance, I need to fiddle a bit with selecting which properties to set on the paragraph, as the paragraph metrics are important for Krita's baseline aligment. Text on Path/Text in Shape. Text on path and text in shape were actually going to be tackled in the first phase. But then when Alvin, who was helping me, tried to take a stab at it, he was blocked because Dmitry was extremely unsure about the design. This was a bit frustrating, because I had already made sure that the layout algorithms for both worked fine. I myself decided to focus on getting rich text editing to work, and it wasn't until I was nearly done with rich text editing that we returned to the discussion about text on path and text in shape. I never talked about text in shape previously, as it was implemented after I wrote the big text-layout blog post back in 2022. So lets do that now: The current look and feel for a complex text-in-shape while editing the text. Yellow ellipse, pink triangle and blue polygon are "Inside" shapes. The green rotated rectangle is a "subtract" shape. Both shape padding and margin are applied. The arrows indicate the order of the flow shapes. Icon on the top right is a button for contour mode. SVG 2 allows wrapping text in shapes, and has some sophisticated toggles to configure how the text is flown into shapes. The simplest is a single text in a single shape with a single subpath. However, those single shapes can handle multiple subpaths (in which case the line is broken up), and there can be multiple shapes that text can flow into, one-by-one (much like CSS columns). These flow areas can in turn have other shapes subtracted from them. Finally, shape padding and margin can be used to modify the distance of the text to the related shapes. Most text-in-shape layout algorithms will do so by taking the shape, drawing line boxes from top to bottom, and fitting the text in the first reasonable line box. This is what SVG 1.2 specified for it's text in shape. For SVG 2 however, the first line needs to sit snugly against the border of the text wrapping area. For this I implemented an algorithm described by Hans Muller, originally devised for CSS shapes. Once we have the first position, we want to create a line box. One problem here is the question "how tall is the line box?", especially in rich text: Text can have different font sizes, and different font sizes can lead to different line widths, depending on whether those differently sized sections get onto the line or are wrapped to the next. The solution here is to estimate the line height: We check all glyphs that might fit in the next shape-bounding-box width (or height for vertical), and get the max line height from this. Then, the line box is determined, with a single line being able to hold multiple line fragments if the shape boundaries cut through the line. Text is then laid out onto these lines (logically if we take bidirectional algorithm into account), breaking where it is allowed. Finally, text is reordered so it lays out visually, text alignment and justification takes place, and the final line height is calculated. The whole text is then shifted (block wise) upwards by the difference between the estimated line height and the actual line height. I was able to figure out this solution because I kept trying to figure out what Inkscape was doing here, and was able to induce a bug that suggested it does something similar. Said bug got reported. Some oddities are present in the SVG 2 spec here. For one, it doesn't ever say whether to include the local transform of the shapes that is being flowed into. Text on path does require this, and usage would

become incredibly annoying without it (if you want to flow text into two rectangles, you will need to apply a transform to one of them), which means it is expected, but its absence is very odd. Inkscape does do this, so I implemented this as well. Another thing that can be odd is that because the shapes get linked to the text shape, it is possible for the text to be rotated and be out-of-sync with the linked shape. The only way to have both rotate together is when they are in the same group. In the context of SVG this linking behaviour makes sense. You can easily imagine a magazine layout where a few rectangles provide columns, and a pull-out quote is laid out into a circle. Said circle then overlaps these rectangles, without overlapping the text. However, while that makes sense in a magazine context, there was a worry that it might be too complex to interact with. Especially because Krita doesn't have an object outliner like Inkscape or other specialized vector applications have, so going in and out of groups can be frustrating. Eventually we decided to make it so that Shapes that text flows into is always a child of said text. The text is then stored as a group with shapes and text inside SVG, making it 100% compatible with Inkscape, while within Krita we could simplify the interaction, while keeping all the powerful transformation features. There's a downside to this method though: When we resize text in shape, we resize the child shapes. However, the child shapes affect the slow text layout (in particular, the shape-offset operations, which are quite slow), but because the child shapes are children of the text, it becomes next to impossible to update this text independently from the resized shapes. Very frustrating because I did pay a lot of mind to keeping the text layout thread-safe, so had we stayed with a linking model, the text shapes could've just been sent to another thread to sort themselves. Thankfully Dmitry decided to take responsibility for taking care of this slowdown. We now block text layout during resizing and update it after the fact. Because of the way bounding rects are calculated however, this does mean we cannot afford to have any text drawn outside the text shapes, which means we're forced to have our overflow to be always clipped (and truthfully, there's no clear answer to what overflowed text should look like with SVG 2 anyway, so maybe it's for the best…). UI wise, the simplest way to set text in shape is by clicking a shape. Clicking on a border will instead set the text on path, and set the click location as the starting point. This is necessary because right-to-left text needs to be aligned with the end of the path due the way text on path interacts with text anchor in SVG. The less simple but advanced manner is to use the context menu in the Shape Selection tool to flow texts in Shape. This method allows for setting up a complex flow structure. Similarly, the default tool allows changing the flow shape order, and setting subtract shapes. Both this and the previous method are ways in which such shapes are set up in other programs, so people should be able to find either without consulting the manual. When a text-in-shape or text-on-path is created, a new button appears that can be clicked to go into contour mode. There, each contour shape can be manipulated as needed. Within the text tool, text on path gains a handle to move the start offset, while text in shape allows dragging the text area to set the shape padding and margin. Text-on-path doesn't have an advanced mode like text-in-shape, right now artists will only be able to create texts with a single text path. However, Krita's text layout can handle multiple text paths in a single text, and even a mix of text and positioned paths. That kind of thing is currently limited to the SVG source editor, as I ran out of time to ensure that the interaction would be nice. Something for the future. Type Setting Mode Type setting mode is a separate mode in the text tool that allows for on-canvas fine tuning and interaction with font metrics. It differs from regular editing mode in that it will show editable font metrics when activated. When the text doesn't auto wrap, it even shows handles so that the SVG character transforms can be modified over the selection. Type Setting Mode is kinda interesting in that at first glance it seems like an unnecessary toy mode. After all, if you want to edit the font size, the text properties docker is much more suited, right? Yet, when gathering input about what artists needed from the text tool, some expressed that they wanted to be able to edit things like font size and line height by on canvas widgets. Others protested: it would interfere with text editing, which seemed a reasonable concern. So it was clear that if such a thing would be introduced, it needed to be optional. Then there was the issue of the

Baseline features. Krita is currently one of the rare text layout implementations that implements alternate baseline alignment. But the baselines are kind of abstract, especially as font makers rarely fill out the OpenType BASE table from which these baselines are derived, meaning they frequently have to be synthesized. So there was also a need to allow people to see the available baselines on canvas. But there was one final issue. Let's talk about kerning. If we conceive of text as being comprised of glyphs, and each glyph can fit onto a little rectangle, like in (movable type) print, and we imagine printing with this. Then it's very likely that there will be huge gaps between the glyphs while printed. Therefore, font makers would make the base rectangle smaller and let parts of the letter overhang, a so-called "kern", so the glyphs would interlock a bit more elegantly. Movable type was never the only text printing technology. For posters for example, lithography was widely used, and the text printed with lithography was typically hand drawn by the artists. This meant that artists would be able to manually decide the best spacing for a given piece of text. Then, there's a number of in-between technologies. There was a particular one where designers would work with letter sheets than could be transferred onto a given piece of paper, and I seem to recall there's a similar technique that relied on clever use of photocopiers. The precise technologies aren't very important here, but rather I want to impress that there's a western practice of spacing glyphs in a text just right, and that the underlying technology greatly affects what is possible. As such, this practice is taught to students of design, and seen as one of the important details that distinguishes a well done piece of typography from a rush job. For the western typographer, to get the kerning just right is to say you care. Now, in the digital era, font makers are able to very quickly define kerns for any pair of glyphs, and while doing text layout the shaper will apply these kerns. This is generally good enough for the majority of use cases… However there's a technological limitation. See, if you do rich text layout, you first need to itemize the text into ranges where the font, direction, and script is the same before you hand it over to the shaper to shape. In the above example, you can see that the first letter is much larger than the rest, and there's no kerning. This is because the font size is different, and thus, during itemization, it's a different font, and a different glyph run. The shaper cannot apply kerning between these two different runs. Typically, this is worked around by adjusting the tracking or kerning. CSS however, only has letter-spacing, which is meant to be applied to ranges of text. Meant. In practice, the majority of implementations make it so that letter-spacing modifies the spacing to the right of clusters of glyphs. But not all: some implementations do it to the left when text is right-to-left! If that weren't enough of a headache, right now, the CSS working group is changing the way letter-spacing works all together. Using letter-spacing for this is not much of an option then. The CSS-WG suggests that if you really want to do manual kerning, you need to create a special span with reduced margins, as this will give the most control. SVG doesn't have margins though, as SVG doesn't have the CSS box-model. But we do have character transforms in SVG. Character transforms in SVG have been there since the beginning. There's 5 parts to character transforms: absolute x and y, relative dx and dy, and rotate. Absolute X and Y set the current text position in absolute coordinates to the text origin, and, these break shaping, much in the way line fragments do with auto wrapped text. When the SVG text specifications talk about text chunks, it means ranges of text that have been positioned this way (and since SVG 2.0, also other forms of line fragments). Dx and dy conversely, accumulate, starting from each text chunk start. They don't break shaping, nor does rotate, which means these three are very suited for this need for manual adjustment. They just needed to be editable. So when I was looking at these three issues (on canvas adjustments, baseline selection and character transforms) and was deciding on my design, it became clear we needed this separate mode to handle these three things, and also that it wasn't all that optional: spacing and kerning are a pretty important practice after all. I then spend 10~ days to get the character transforms right. This was because I decide it would be more useful to calculate the relative positioning from absolute positions rather than to set the relative positions directly. This way, I would only need to calculate where I want the glyph to be, and let the function itself sort what kind of delta positioning that requires. This required me to

kind of backtrack from the final position to calculate the point at which the delta x and y are added. This was quite tricky as there's a lot of modifiers on where a particular transform ends (ligatures, utf32 vs utf16 codepoints, and of course, white space collapse), as well as going backwards, as that involves removing the text-path adjustment, text anchor calculation, absolute offset and finally the textLength offset. The actual editing of this is provided by two handles at either side of the active selection. Dragging the square handle offsets the whole selection, while dragging the round handle scales and rotates the selection, using the square one as the hinge/origin. I am not fully sold on how this is handled, especially in RTL, and I want to see if I can handle the offsetting better. With type setting mode, you can transform each glyph individually, allowing for some pretty advanced looking typography. The lines are the metric lines. Hovered line is the Descender, hence it saying "Font Size". Next up was changing on canvas properties. This is right now, limited to Font Size, Baseline Shift and Line Height. The baseline shift is modified by dragging the baseline, the font size is modified by dragging either the ascender or descender, and the line height by moving the line height markers, which are ascender and descender + the line-height on either side. Artists will be able to tell which they're modifying by the hovering name. Krita slogan in Hindi. Here, we use type setting mode to adjust the size of the second word, and then select the "hanging" baseline to align the text to the head stroke. Setting the dominant/adjustment baseline can be done by pressing Shift, which switches the visible lines to the baselines. Clicking them will set that as the dominant baseline. There's still an issue here with overlapping lines, and I need to sit down and think about which lines should have priority. This is pretty useful already, but there's still a number of unanswered questions: Right now, I add a counter transform at the end, this is because when using it, the counter transform felt more intuitive. However, it can also make sense to not have that. Maybe it needs a toggle? Similarly, the scaling/rotating code can easily only do scaling OR rotating, and it makes sense to use either. But I am unsure how to provide that in the UI. Thirdly, there's right now no way to set the Absolute transform. I got some functioning unittests for it, but some edge cases look weird and it needs more work before I can expose it to the UI. Right now, the Font Size, Baseline Shift and Line Height are all adjusted in Points. I was unsure whether to have them in relative font size, and we'll need to see if that's something people prefer. I can imagine that some people would like to see a line over the x-height, but the thing is that there's no real related metric for that. There's font-size adjust, of course, but newer versions of font-size-adjust are also possible against the capital height, or even the ideographic height. So I just left it out for now. The metrics that can be adjusted are all of a certain type, what western Typographers call "vertical metrics". There's no controls yet for Tab size, text indent, word spacing and letter spacing, though they could be easily imagined. In a similar vein, one could imagine handles for italic/slant, weight or width. These I have been avoiding because of these, only slant can be predicted, the other two are unique to the type face. Finally, there's of course associated shortcuts. I've implemented four for moving the offset in any of the four directions, but none for scaling/rotating. This is because I assumed people would definitely want to offset with the keyboard, but was unsure about the others. Part of these are because Type Setting Mode came in very late. It was always going to be added as last, because, from a surface level it sounds like a frivolous toy mode. When I expressed my intent to create it, some artists even told me they were never going to use it. Not strange: There has been over 30~ years of digital type setting that didn't need a separate type setting mode. But once you see the whole picture, and more specifically, realize that not all text layout systems are the same, the purpose starts to make a little bit more sense. Whats more, because SVG character transforms have been there since the beginning of the SVG spec, they're pretty widely supported, so it'll be very interesting to see what people will come up with. There's still a snag though: SVG relative character transforms don't apply on auto-wrapped text. There's a little note in the SVG 2.0 spec that these were considered, but ultimately seen as unnecessary. Little bit annoying, but not the end of the world, as someone who aims to wrap in shape, but then fine tune, can do just that. Wrap in shape, convert to pre-positioned text, and fine

tune the spacing… Let’s now finally talk about the converters. Conversion actions for text types. Because previous versions of Krita only supported SVG 1.1 text, it was important that there would be a way for people to convert away from that format. Similarly, if someone had put text-in-shape, or created an Inline wrapping area without intending to, there needed to be a quick way to convert. Converting away from “Pre-Positioned Text” (SVG 1.1 text with white space collapse), required first removing all collapsed white spaces. Then, inserting new lines for each SVG text chunk with absolute positioning. All of this needed to be done in reverse because insertion and deletion changes the indices. By going in reverse the indices that still needed to be modified were kept the same until modification. For inline size, the inline-width of the text is tested before conversion, and set after conversion. Converting towards Pre-Positioned text on the other hand was much easier, as it was a case of figuring out the current position and making sure it was being set as an absolute transform. Because we are just working with the layout results, we can convert from text-in-shape to pre-positioned and keep the lines positioned the same. The leftmost polygon here is a text in shape, while at the right, the text has been converted to pre positioned SVG 1.1 text. The difference is only visible because of the different selection rectangles. These actions were implemented in the text tool for single shapes and in the shape select tool for multiple ones. Wrap up And that was all of Phase 3, which means I am done for Krita 5.3. As of writing, we’re in feature freeze. This means I will be focusing on fixing bugs in the coming few months. But also writing documentation, and release notes. I am slightly worried, as I didn’t get a lot of feedback near the end, and am left wondering what kind of bugs I will see in the coming months. Some things didn’t get in from the original plan. Most notably: color and stroke setting. Krita can set these things, the controls for it need to be ported to QML, and I was told to avoid it, as it would be too big. Due these missing controls, stroke can only be set for the whole text, while the fill can only be a color for a selection, or gradient set on the full text. Beyond that, a visibility mode for formatting marks (that show where the spaces are and what type of spaces they are) didn’t get in either. It’s by itself not complex to implement, but it needs good design of the marks, and I just didn’t feel like I could give it the attention it deserves because of the amount of work that went into text in shape. It is also not yet a full implementation of SVG 2. Text-orientation is the biggest missing element here, but I told myself I wasn’t going to work on that until there was a decent enough text editor. There’s also things like better justification, hyphenation and color font support, but those were never going to be in 5.3. It’s going to be interesting to see how the usability is going to be tweaked over the coming years. A good number of properties can be found directly on the canvas and the main dockers, so I do feel everything is pretty discoverable. However, I did have to put the advanced text-in-shape actions into a right click menu, and generally people don’t find those. It will also be interesting to see what people will do with the type setting mode and how that’ll evolve. I know people want an on canvas property editor, but I had been holding off on that because the design would be tricky to get right, as well, in Qt5 Android platform integration doesn’t yet obey the enum value that asks for the copy-paste menu to be hidden. So that menu would probably float over any on-canvas property menu. Qt6 fixes this, but I don’t know if it does for Apple products as well. Overall, this was a pretty ambitious project. One thing that probably didn’t bleed through in all these blog posts is that probably at a least a third of the work was communication. I myself understood that when I started it, but I think it has a tendency to get lost when you read tech blogs, so I’ll expand a little about it. Basically, every step I took, I spend some time talking with the other Krita developers (primarily Dmitry, who reviewed all my work) what I wanted to do, and how I was going to approach it. This isn’t just to get the design sorted, but also to avoid blind-siding people. I also wrote these blog posts, and wrote little feature introductions for the people on Krita artists. The purpose of the latter was to get people to understand with what I am trying to do, I am unsure how successful I was there. The technical blogposts did appear to be pretty helpful, and I got a lot of feedback from other FOSS people that these blog posts were useful. While text layout is not a rare programming topic, advanced text layout is only really done by a handful of people, and I imagine a lot of

them are too exhausted after they got the thing to work to write a blog post about it. Anyway, when I first started, my colleague Agatha had described text layout like a “Hydra” and that every update I made felt like I was chopping those heads off one-by-one. I declare the Hydra dead: Krita has a decent text tool now. Appendix SVG Character Transforms. Just checking how widely supported the SVG character transforms are… Inkscape and Chromium do pretty well. Firefox does a little bit odd. SVG Tiny 1.2 does include the character transforms, but QtSvg doesn’t support them (or Gwenview isn’t using QtSVG). Then there’s a lot of epub readers that have varying degrees of support. The sample SVG-based epub3 files do use multiple character transforms, but that by itself doesn’t mean much. KoReader for example only supports the first transform, even with text-on-path, which is interesting given that LunaSVG, which it uses, does seem to have support for it if you look at the code. KoReader does apply textLength, which is useful, but then other epub readers I’ve tried don’t apply textLength, but only transforms. It’s a bit of a mixed bag, but the wide browser support is heartening (good enough if you just want to tweak spacing while maintaining an accessible SVG text element). Our “The Two Towers” sample in Firefox 145. Firefox doesn’t support baseline shift (but does support white-space, so we can preserve the line break… but then doesn’t support SVG units, so all units need to be suffixed with px) Chromium 142, it supports SVG 1.1 fully, but no white space, so I had to convert to pre-positioned text for this to work. Neither browser supports optical size in SVG, so the large capitals are less delicate than in Krita. The sample in Inkscape 1.2, I had to convert the font from Amstelvar to DejaVu Serif, as it didn’t seem to like Amstelvar (probably because it is a variable font). It’s by far more widely supported than SVG 2 text wrapping though, of which the only known (to me) implementations are Inkscape and Krita. Ideally, Krita, like Inkscape, would ensure there’s fallback positioning written. The absolute transforms are 100% intended to provide a fallback when the auto wrapping is not yet supported. The reason Krita isn’t doing this is because it needs a separate code path so it only saves this to exported SVG layers, and as well, I am not 100% confident in my conversion code. When that time comes it might also prove useful to save the textLength, but I’m still mulling over this. A final note is that Krita, like Inkscape, has a “convert text to paths” function. When converted to paths, text can be modified as desired, but then it loses the accessibility of being text that can be selected, which you’d want to avoid in an interactive environment.

- From Commodity Trap to Sustainable Innovation: Convincing Automotive Executives to Embrace Open Source (2025/11/19 06:00)
  Automotive companies spend too much on commodity software. Co-creating multiplies their investment through community collaboration. This frees resources for true differentiation. Read about, the concepts and ideas to master, that support this strategic journey.
- Design System Progress – November 2025 (2025/11/18 00:13)
  For the past few weeks, we have been working on a few areas around the design system for Plasma. Keep in mind that I am only speaking of the graphic side, not code. Work is ongoing with the Union engine and the team is focused on replicating our current Breeze style using Union. There have been talks about creating the first components based on the design system, but that is more in the future. Meeting with PenPot We held a meeting with the PenPot team and Pablo Ruiz, their CEO, met with us to discuss new changes in the PenPot app. This was a follow up to their recent conference PenPot Fest. Their team announced a few things that should make it much easier for the Plasma Design team to adopt PenPot. For example: New composite token: Typography Taiga #10200 Show current Penpot version Taiga #11603 Switch several variant copies at the same time Taiga #11411 Invitations management improvements Taiga #3479 Alternative ways of creating variants – Button Viewport Taiga #11931 Reorder properties for a component Taiga #10225 File Data storage layout refactor Github #7345 Make several queries optimization on comment threads Github #7506 With these additions, it was much easier to move assets into PenPot than before. There was less work we needed to do. We begun a migration to PenPot for the second layer of basic components and also started building more complex components.

Here are some screenshots: Buttons Button Groups Badges Inputs Dropdowns Toggles Checkboxes Checkbox Groups Avatars Tooltips Progress Indicators Sliders These components are shared components. Then we moved into application components and this is what we have so far. Application Components Modals Pagination Tables Video screen (Miscellaneous) Breadcrumb Tabs Alerts and Notifications Date Pickers File Upload Section Headers Content Dividers In this list, you see a lot of graphics. Each of these is supposed to represent a different state of the graphic. Users wouldn't work with these variant sets very much, instead, they would simply search in the component catalog for what works in their design and only edit organization and labels. However, to get to that level, the designers need to create interpretations of each of these states graphically. This leads to a lot of work and a lot of graphic memory usage. There are a few more components that can be created. However, given PenPot's reliance on the browser DOM, the more complex the components, the more lag the application runs into. Because of this issue, we have contacted PenPot to become beta testers of their new rendering engine when it comes out. They are almost at the point where they can put this out. We are eager to try and see how much faster we can go. The issue is not on PenPot but the engine that powers the editing screen. Still, we have to wait a little bit to continue. In the mean time, we can dedicate ourselves to making more application icons and completing the work there. FOSDEM Additionally, we are setting up a workshop with the PenPot team during FOSDEM 2026. This workshop will focus on brainstorming ideas on how to more easily distribute and contribute to a design system using PenPot. For example, there is a list of ideas we proposed: Exclusions and inclusions into the design system library. This way, the original copy of the design system remains consistent with the base components unalterable. This should make it easier for casual designers looking to build a quick mockup without getting bogged down by sub components that don't need edits. This can also ensure that the many users taking the components are using a consistent copy to the original. Automatic sharing and updating to users not in the immediate instance team. Generate a review system for components as external users to the main instance propose changes. An easy way to re-publish the design system after applying suggested changes. …and a few other ideas. Hopefully, there are good ways to get this done. We are still waiting to move our icons into PenPot. Likely, this is more of a reality once the new rendering engine is in place. The team let us know that there are a number of shape manipulation improvements app In addition to all of these changes, we keep submitting bug reports and feature requests to the PenPot team to make the app even stronger.

- KDE Plasma 6.5.3, Bugfix Release for November (2025/11/18 00:00)
  Tuesday, 18 November 2025. Today KDE releases a bugfix update to KDE Plasma 6, versioned 6.5.3. Plasma 6.5 was released in October 2025 with many feature refinements and new modules to complete the desktop experience. This release adds two weeks' worth of new translations and fixes from KDE's contributors. The bugfixes are typically small but important and include: View full changelog
- This Week in KDE Apps (2025/11/17 22:10)
  Crop tool in Photos, Sudoku in Kirigami and sprintingWelcome to a new issue of "This Week in KDE Apps"! Every week (or so), we cover as much as possible of what's happening in the world of KDE apps. Last Saturday a bunch of KDE devs (and a guest) met in my kitchen for a "Kitchen sprint". As always, we discussed and worked on quite some exciting stuff, mostly around Itinerary and public transport infrastructure in KDE, but not only. Here is a short overview of what some of us worked on: Jonah experimented with integrating maplibre in our apps, Nico demoed his new online account integration for applications, and, outside of cooking some Käsespätzle for the whole group, I spent some time packaging Merkuro as a flatpak! Outside of that, and as part of our end-of-the-year fundraiser, you can adopt one of KDE's apps and we can share with the whole world how awesome you are and how much you're doing to support us. Thanks to everyone who already donated, this is super helpful! Getting back to all that's new in the KDE app scene, let's dig in! Multimedia/Graphics Applications Photos Image Gallery Noah Davis added a crop tool to

the image editor of Photos. (25.12.0 - link). Joshua Goins improved the performance a bit in the main view (25.12.0 - link). Sytem Applications Dolphin Manage your files Nate Graham reverted a change which impacted keyboard-driven folder manipulation (25.12.0 - link). Oliver Schramm fixed trashing files from temporary folders. Now they no longer end up in your home trash bin. (KDE Frameworks 6.22 - link) PIM Applications Merkuro Calendar Manage your tasks and events with speed and ease Tobias Fella fixed setting the calendar name (25.12.0 - link). He also disabled the calendar editor when we don't have permission for it (25.12.0 - link). Social Applications NeoChat Chat on Matrix Tobias Fella simplified the process to unlock the key backup by providing only one text field (26.04.0 - link) and it is no longer behind a feature flag (link). Tokodon Browse the Fediverse Loïs Rioul fixed login with GoToSocial (25.12.0 - link). Games Pumoku Anders Lund pushed the first early alpha version of his Kirigami based sudoku application called Pumoku. It is still a bit basic but very promising. Third-Party Applications Easy Effects - Audio Effects for PipeWire Applications Wellington Wallace released Easy Effects 8.0.3 containing a bunch of fixes for regression from the major 8.0.0 release. Giusy Digital fixed some translations issues in the spinboxes (link) and the number validator (link) Carl Schwan ported the settings to KirigamiAddons ConfigurationView (link) Carl also fixed various spacing issues in the effect pages (link), ported the navigation menus to normal tool buttons (link), ported the application metadata to KAboutData and FormCard.AboutPage (link) and various other small graphical changes. …And Everything Else This blog only covers the tip of the iceberg! If you’re hungry for more, check out Nate's blog about Plasma and be sure not to miss his This Week in Plasma series, where every Saturday he covers all the work being put into KDE's Plasma desktop environment. For a complete overview of what's going on, visit KDE's Planet, where you can find all KDE news unfiltered directly from our contributors. Get Involved The KDE organization has become important in the world, and your time and contributions have helped us get there. As we grow, we're going to need your support for KDE to become sustainable. You can help KDE by becoming an active community member and getting involved. 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. There are many things you can do: you can help hunt and confirm bugs, even maybe solve them; contribute designs for wallpapers, web pages, icons and app interfaces; translate messages and menu items into your own language; promote KDE in your local community; and a ton more things. You can also help us by donating. Any monetary contribution, however small, will help us cover operational costs, salaries, travel expenses for contributors and in general just keep KDE bringing Free Software to the world. To get your application mentioned here, please ping us in invent or in Matrix.

- A new online accounts system? (2025/11/15 09:00)
  For many years Plasma comes with its own system online accounts system, known as KAccounts. The idea is simple: In Systemsettings you log into a given online service once, and then several applications can use that login, instead of loggin in inside each application separately. The number of services available and applications making use of them changed a bit over recent years. As of right now the following services are supported: Nextcloud: This is used by Dolphin to add a shortcut for file access via webdav to the Network section, as well as the Purpose framework to allow uploading files to Nextcloud. Owncloud: Used for the same things as Nextcloud. Google: Used by Purpose for uploading to YouTube. In theory also used by kio-gdrive for browsing Google Drive, but access to this is currently blocked by Google. OpenDesktop: Used for reviewing store.kde.org content in Discover This isn’t all that much. Notably absent here is KDE PIM, which could greatly benefit from integrating with the Nextcloud and Google accounts. This is something many users have asked for over time. Plus, there’s more services that are used across applications and could benefit from a systemwide online accounts system, like Mastodon or Matrix. Overall the situation with online accounts support in KDE is unsatisfactory, and it’s not for a lack of trying. Over the last few years there have been several smaller improvements

to the system. However there have been no changes to the overall architecture. At this point I am convinced that the architecture is what's holding us back, and we need to do something about that. The current system is based on the Accounts SSO framework. It consists of several libraries and processes, split across about a dozen different repositories. This makes for a rather complex system for what is effectively reading and writing to a sqlite database and some OAuth handling. It also receives very little development activity, to the point where it was hard to get the required patches for Qt6 support in. Using an external accounts system as based for KAccounts only makes it harder to iterate on our system, while providing no meaningful interoperability with other parties. The system also isn't designed for a sandboxed world. Apps have direct access to the accounts database and keychain, so there is no ability to restrict which apps can use which accounts. While per-application access control wasn't really feasible for traditional Linux packaging, with sandboxed formats like Flatpak we can and want to restrict apps to only be able to access select accounts. Having pondered the problem for a while I came to the conclusion that we need a fresh start for our online accounts story, a new system that fullfils the following goals: It's actually used by relevant KDE and third-party software Easy to hack on and extend Easy to be integrated into consumer software, with minimal dependencies Can be extended with third-party providers Account data is stored reasonably securely, with per-application access control (for sandboxed applications at least) Based on these goals I have developed a prototype for how such a system could look like. At its core there is a daemon process that manages the accounts, and exposes them via a DBus interface. Applications uses this DBus interface to list availble accounts as well as their parameters and credentials. Only accounts the app has been granted access to are visible that way. Application authentication works in a way that's inspired by how xdg-desktop-portal works. An application can trigger a request for accessing a new account. The dameon will then handle the whole login flow and, if successful, will return a handle to the new account. Alternatively the user can log into a given service in the systemsettings module and give access to relevant apps though that. Currently the following services/apps are supported: Nextcloud: Used by Purpose and KDE PIM Mastodon: Used by Tokodon Google: Used by KDE PIM and Purpose You can find the code at https://invent.kde.org/nicolasfella/konlineaccounts. It is still very much a prototype, which is by no means ready for production, but it shows the basic concept. If you have input on this please get in touch, for example by filing an issue.

- OSM Hack Weekends October and November 2025 (2025/11/15 07:15)
  Last weekend I once again attended the bi-annual OSM Hack Weekend in Karlsruhe hosted by Geofabrik. I've also been at the OSM Hack Weekend in Berlin hosted at Wikimedia Deutschland a couple of weeks ago and haven't written about that yet, so this is the combined report for both events. Transitous Transitous, our community-run public transport routing service, has been the topic of several discussions: Ways to deal with GTFS static feed rotation happening out of sync with corresponding realtime feeds. This results in time periods where available realtime information cannot be matched to base schedule data and thus gets needlessly discarded. How to best configure GBFS provider groups as supported by MOTIS v2.7. Integrating GBFS data from Citybikes, which would substantially increase the amount of available rental vehicle data. Assessing what it would take to add Transitous as an additional routing option to the OpenStreetMap website. Investigating how far along the OSM Road Closures GSoC project is, as that kind of data is obviously very interesting to integrate eventually. Exploring whether FOSSGIS e.V. would be a suitable organisational home for Transitous. Available rental vehicles shown on Transitous' map view. KPublicTransport KPublicTransport, KDE's client library for accessing different journey planning services used by Itinerary and KTrip, got a few improvements to catch up with Transitous and MOTIS v2.6 and v2.7 changes: Access to agency/operator URLs. Querying available station-bound and free-floating rental vehicles from MOTIS. Support for direct booking URLs for station-based rental vehicles. Station-bound rental vehicles other than bikes are now also displayed with the correct vehicle icon on the map. Itinerary's station map showing a car rental station and two free-floating rental

bikes. Indoor mapping Indoor mapping was of course also on the agenda: I got to try Tobias's JOSM patches improving level filtering. Especially the option to filter on elements without a level tag is helpful for fixing level tagging in existing buildings for me. We talked about ongoing tagging discussions from TU Munich's BIM import, in preparation for the next quarterly OSM Indoor Meetup. We discussed whether we should have another in-person Indoor tagging workshop following the one from 2022, in order to have some time to work on finalizing tagging proposals and updating the current indoor tagging documentation. Emergency and weather alerts At the CAP Implementation Workshop two weeks ago a WFS/OGC feature layer for CAP alerts was mentioned, and presented as something so far only offered by a commercial entity. With my almost non-existent GIS knowledge this looked like something that shouldn't be too hard to provide by our CAP alert aggregation service as well. And thanks to the input from the right people I got a basic prototype set up in less than an hour. All the magic is provided by pg_featureserv, which can expose a PostGIS database (which we already have) in a way it can be consumed by e.g. QGIS. QGIS with a CAP alert message layer. One important difference here is that unlike its proprietary counter-part this doesn't expose many CAP fields yet, as we hold only the bare minimum as dedicated database columns right now. However, should anyone actually need this, adding more columns isn't a big deal. Event planning We also looked at upcoming events in 2026 and how we could have Transitous specifically and the Open Transport community more generally represented there: 39C3, 27-30 Dec in Hamburg Germany. We'll try to have some kind of Transitous meetup there. FOSDEM, 31 Jan-1 Feb in Brussels, Belgium. The CfP for the Railways & Open Transport track is still open and we have poked a few people to submit talks. FOSSGIS-Konferenz, 25-28 Mar in Göttingen, Germany. The CfP is already closed, a few proposals have been submitted. Still further out is next year's State of the Map which will be end of August in Paris, France. That's obviously something where Transitous should be present as well, and where we might have the option of a travel-optimized adjacent Transitous sprint along the way. Ideas for a 2026 edition of the Open Transport Community Conference are also floating around already, volunteers to drive this still very much needed though. You can help! Hack weekends how this is called in the OSM community or sprints as this is known in the KDE community are immensely valuable and productive. There's a great deal of knowledge transfer happening, and they are a big motivational boost. However, physical meetings incur costs, and that's where your donations help! KDE e.V. and local OSM chapters like the FOSSGIS e.V. support these activities.

- This Week in Plasma: OCR in Spectacle and many UI improvements (2025/11/15 00:01)
  Welcome to a new issue of This Week in Plasma! This week Spectacle gained OCR (optical character recognition) functionality, allowing you to turn words in images into selectable text! (and yes, this is Spectacle recording itself performing OCR) Right now the functionality is limited to Spectacle, but the code is in the process of being moved to a library so more apps can benefit, too. Thanks a lot to Jhair Paris, who implemented this feature that will appear in Plasma 6.6! In addition, many UI improvements landed, as well as some high-priority bug fixes and performance improvements. A good week, I'd say! Have a look: Notable UI Improvements Plasma 6.5.3 You can now drag a tab out of a Chromium/Chrome window and immediately tile it to a screen edge or corner. (David Redondo, link) Implemented some improvements to the Breeze theming for GTK 4 apps, including making the rounded corners consistent and fixing invisible expander arrows for expandable group boxes. (Kevin Duan, link 1 and link 2) Made the favorites column in the Kicker Application Menu widget compatible with more kinds of icons in non-default icon themes. (Christoph Wolk, link) Plasma 6.6.0 Overhauled a bunch of the portal-based permission dialogs to just look way nicer in general. (Harald Sitter, link 1, link 2, link 3, link 4, link 5, link 6, link 7, link 8, link 9, link 10, link 11, link 12) Renaming a file or folder on the desktop now lets it keep its existing position. (Błażej Szczygieł, link) Plasma's Timer widget now has a nicer and more straightforward configuration page, with two old pages merged into one new one. (Tobias Fella, link) For similar reasons, the system Tray's only two main configuration pages have also been merged

into a single new one. (Nate Graham, link 1 and link 2) When an app asks to register keyboard shortcuts on launch and you don't let it, this preference is now remembered, instead of the annoying app just asking again the next time it launches. (David Redondo, link) The GTK theme chooser now lets you preview the dark version of the theme, too. (Luan Oliveira, link) Frameworks 6.21 File transfer notifications now fall back to file-based progress display in situations when size-based progress display isn't available. (Pan Zhang, link) The Breeze icon theme now includes icons for Nim code files. (Sophie Ahumada, link) Notable Bug Fixes Plasma 6.4.6 Fixed a seemingly random Plasma crash. (David Redondo, link) Plasma 6.5.3 Fixed a regression that made KRunner crash when searching on operating systems that use Musl as their C standard library instead of Glibc. (John Zimmermann, link) Fixed a case where Discover could crash while updating software if Flatpak itself throws a malformed error. (Aleix Pol Gonzalez, link) Fixed a regression that broke Spectacle's "exclude shadows" option. (Vlad Zahorodnii, link) Fixed a regression that made desktop icons not get visually hovered when approaching them from the left side at certain speeds and locations. (Błażej Szczygieł, link) Fixed an issue that made the context menu for apps or processes being monitored in System Monitor open in the wrong place when using certain multi-screen setups. (Oliver Schramm, link) Worked around a Qt regression that broke hiding columns in System Monitor's table views using the column header context menu. (Alexey Rochev, link) Worked around a kernel bug that that made some systems fail to go to sleep the first time it was initiated. (Bhushan Shah, link) Worked around some driver bugs that caused mangled cursor styling with certain GPUs. (Xaver Hugl, link 1 and link 2) Other bug information of note: 4 very high priority Plasma bugs (same as last week). Current list of bugs 31 15-minute Plasma bugs (down from 35 last week). Current list of bugs Notable in Performance & Technical Plasma 6.5.3 Fixed a case where KWin could get blocked due to heavy disk I/O operations. (Vlad Zahorodnii, link) The clipboard portal now supports being used in remote desktop sessions and with middle-click primary selection. (David Redondo, link 1 and link 2) Increased the level of visual fidelity when using a fractional scale factor. (Xaver Hugl, link) Plasma 6.6.0 Increased the level of visual fidelity when using a fractional scale factor even more, this time when using software rendering. (Vlad Zahorodnii, link) A process that's crashing in a loop can no longer make the system run out of memory and freeze as a result of the crash tracer trying to debug all the crashes. (Harald Sitter, link) How You Can Help Donate to KDE's 2025 fundraiser! It really makes a big difference. Believe it or not, we've already hit out our €75k stretch goal! I've been informed that a second stretch goal is available now, too! I'm just in awe of the generosity of the KDE community and userbase. Thank you all for helping KDE to grow and prosper. If money is tight, you can help KDE by directly getting involved. 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. To get a new Plasma feature or a bugfix mentioned here, feel free to push a commit to the relevant merge request on invent.kde.org.

- Web Review, Week 2025-46 (2025/11/14 12:04)
  Let's go for my web review for the week 2025-46. Valve is about to win the console generation Tags: tech, valve, kde, hardware, gaming, desktop Early days, we'll need to see the pricing and reviews. I'm obviously excited to see KDE going in even more consumer devices by default. Let's hope it sells even better than the Steam Deck. https://xeiaso.net/blog/2025/valve-is-about-to-win-the-console-generation/ Meta's Chief AI Scientist Yann LeCun To Depart And Launch AI Start-Up Focused On 'World Models' Tags: tech, business, ai, machine-learning, gpt I was actually wondering when this would happen. Was just a matter of time, would have expected this move a couple of months ago. https://www.nasdaq.com/articles/metas-chief-ai-scientist-yann-lecun-depart-and-launch-ai-start-focused-world-models Less is More: Recursive Reasoning with Tiny Networks Tags: tech, ai, machine-learning, reasoning, research Clearly needs further exploration. I'd like to see it submitted in a peer reviewed journal but maybe that will come. Still it's nice to see people for new approaches. It's a breath of fresh air. I like it when there

are actual research rather than hype. Hopefully the days of the “scale it up and magic will happen” are counted. https://arxiv.org/abs/2510.04871 Introduction to IncusOS Tags: tech, virtualization, containers, infrastructure, linux, incus I didn’t know about this project. This sounds interesting, smart use of mkosi to make an Incus tailored system. https://linuxcontainers.org/incus-os/ Linux process priorities demystified Tags: tech, linux, kernel, processes, multithreading Does a nice job explaining how the scheduling can be investigated from outside the kernel. It is a good introduction on the topic. https://www.sigma-star.at/blog/2022/02/linux-proc-prios/ Announcing Magika 1.0: now faster, smarter, and rebuilt in Rust Tags: tech, filesystem, foss Looks more and more like an interesting solution for file type detection. https://opensource.googleblog.com/2025/11/announcing-magika-10-now-faster-smarter.html?m=1 The (lazy) Git UI You Didn’t Know You Need Tags: tech, git, version-control, tools Nice tour of LazyGit. I keep hearing good things about it, I should really try it. https://www.bwplotka.dev/2025/lazygit/ Imperative to relational Tags: tech, databases, sql, learning Long but nice post about all the things you need to figure out about working with databases when the only thing you know is imperative languages. https://madhadron.com/imperative_to_relational.html Automated Equality Checks in C++ with Reflection (C++26) Tags: tech, c++, metaprogramming Another nice use of the upcoming C++ reflection feature. https://lemire.me/blog/2025/11/09/automated-equality-checks-in-c-with-reflection-c26/ Hardening the C++ Standard Library at massive scale Tags: tech, c++, safety, security, standard Interesting work from Apple and Google to have better hardening in libc++. It’s nice to see it ripples through the upcoming C++26 standard as well. https://queue.acm.org/detail.cfm?id=3773097 channels-console: Real-time monitoring, metrics and logs for Rust channels Tags: tech, rust, debugging, distributed, multithreading Early days but this looks like interesting tooling to inspect and debug programs using Rust channels. https://github.com/pawurb/channels-console The state of the Rust dependency ecosystem Tags: tech, rust, ecosystem, supply-chain Interesting analysis of the crates ecosystem. It shows quite well some of the challenges and weaknesses. Nothing to worry about yet about the ecosystem health overall. Still, you should probably be careful when picking dependencies. https://00f.net/2025/10/17/state-of-the-rust-ecosystem/ GPU Glossary Tags: tech, gpu, hardware, learning Looks heavy on the NVidia specifics but it looks like a very comprehensive view of the important concepts in a GPU. https://modal.com/gpu-glossary The “Dependency Cutout” Workflow Pattern Tags: tech, foss, economics, supply-chain, community This is indeed the best way to handle your open source dependencies. I got concerns about the ability to sell that to management though because of the extra steps. It’s also probably why you want to have an OSPO in your company, it’s a good way to lower the barrier for developers to contribute this way. https://blog.glyph.im/2025/11/dependency-cutout-workflow-pattern.html 100% coverage is not that trivial Tags: tech, tests, tdd, coverage In a large codebase it’s not a given indeed. That’s why you want integration tests to get there. https://blog.ploeh.dk/2025/11/10/100-coverage-is-not-that-trivial/ Composable Tests Tags: tech, tests, tdd This is maybe the property of tests which is the most easily misunderstood. It’s not always easy to respect it as well. https://tidyfirst.substack.com/p/composable-tests Forget Chatbots. You Need a Notebook. Tags: tech, note-taking, science, research, productivity I had a few moment like this in my life. I definitely recommend it. I’ve never been more productive than isolated in a mountain with only books, notebooks and pens. https://calnewport.com/forget-chatbots-you-need-a-notebook/ Bye for now!

- KDE Ships Frameworks 6.20.0 (2025/11/14 00:00)
  Friday, 14 November 2025 KDE today announces the release of KDE Frameworks 6.20.0. 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 Attica Use QCOMPARE vs QVERIFY.

Commit. Baloo Use better way to disable session management. Commit. Use QCOMPARE vs QVERIFY. Commit. [app] Skip application/mbox files greater then 10MB. Commit. See bug #443547. See bug #447681. See bug #460882 [balooctl] Writeback config changes before enabling or disabling indexing. Commit. See bug #419708 Bluez Qt Rfkill: Write to rfkill in a thread. Commit. Breeze Icons Fix up "starred-*" icons. Commit. Fixes bug #511152 Add RTL versions of open-link icon. Commit. Fixes bug #506268 Provide external-link-symbolic as an alias to open-link-symbolic. Commit. Try to avoid to depend on stuff generated by CMake. Commit. Add 16px version of the RTL document-send icon. Commit. See bug #509254 Add document-send RTL version. Commit. Fixes bug #509254 Make only 24 px generation a parameter. Commit. Try to avoid that we have targets that run always. Commit. Extra CMake Modules Generate-fastlane-metadata.py use https://kde.org/donate. Commit. Strip username and password from Fastlane source URLs. Commit. KDEClangFormat: Avoid error w/ CMP0175 enabled. Commit. [FindFFmpeg] Skip version check on Windows. Commit. KDE*CompilerSettings: add note why no cmake_policy is set. Commit. Public KDE-specific modules: set module cmake_policy. Commit. Module templates: set module cmake_policy. Commit. Public general modules: set module cmake_policy. Commit. Public find modules: set module cmake_policy. Commit. ECMGenerateExportHeader: set module cmake_policy, not just one policy. Commit. ECMGenerateExportHeader: do not push/pop module policy explicitly. Commit. ECMQmlLoader: generic languages should have lower precedence. Commit. Fixes bug #509902 FindFFmpeg: add AVFILTER and SWRESAMPLE components. Commit. Add FindFFmpeg.cmake used by many KDE project. Commit. Fix QML modules to be rebuilt all the time with Qt5. Commit. Ecm_generate_headers: support headers in clang vs. deduplicated header files. Commit. CMakeLists.txt - remove trailing '/' to fix CMP0177 warning. Commit. ECMSetupQtPluginMacroNames: fix setting up names for own build. Commit. Framework Integration Kpackage-install-handlers: check file.open result. Commit. KArchive Use CamelCase Qt include. Commit. 7z: Change for{if{}} to if{for{}}. Commit. Autotests/kcompressiondevicetest.cpp spell check. Commit. 7z: Fix infinite loop in malformed file. Commit. Ktar: Fix/Tweak the skip error condition. Commit. Add nodiscard attribute to open(...) declarations. Commit. Ktar: Optimize memory allocations. Commit. KTar: Use skip instead of seek for skipping. Commit. TestTarReadWrite: Also test with KCompressionDevice. Commit. Add some const. Commit. 7z: Fix assert/crash in malformed file. Commit. Explicitly convert enum to int for QString::arg. Commit. 7z: Break early on failure in K7ZipPrivate::folderItem. Commit. KBookmarks KBookmarkGroupTraverser: Fix typo in API documentation. Commit. KBookmark::updateAccessMetadata: Utilize QDateTime method that is significantly faster. Commit. KCMUtils Added name sorting to kcmshell6 --list output. Commit. Fixes bug #509050 KCModuleQml: Also force height in addition to width. Commit. GridDelegate & GridViewInternal: Drop mobile tweaks. Commit. KCodecs Allow to also RFC 2047-encode reserved characters. Commit. KColorScheme Fix colors in kdeglobals not being respected when platformtheme is not kde. Commit. KColorScheme: Add FrameContrast API. Commit. KCompletion Use QCOMPARE vs QVERIFY. Commit. KConfig Kconfig_compiler: prevent empty private: section in headers. Commit. Use QLockFile.tryLock with timeout rather then lock forever. Commit. Fixes bug #508253 Add 59ca348606 to .git-blame-ignore-revs. Commit. Kdesktopfileaction: Improve docs strings. Commit. Use QCOMPARE vs QVERIFY. Commit. Typo--. Commit. KConfigWidgets Use QCOMPARE vs QVERIFY. Commit. KContacts Use default. Commit. Add deathdate in vcardtool. Commit. Add Q_PROPERTY. Commit. Add DeathDate vcard4 support. Commit. See bug #337759 Minor optimization. Commit. Add missing [[nodiscard]]. Commit. Add [[nodiscard]] ignore commit. Commit. Modernize use "[[nodiscard]]". Commit. Allow to translate resource type. Commit. Fix use QVERIFY/QCOMPARE. Commit. Prepare to implement sound type support. Commit. Version is used here. Commit. We need to implement vcard4 sound support. Commit. This check was already done some line ago. Commit. Type can be lower case. Fix import emails. Commit. Remove namespace. Commit. Fix some cppcheck warnings. Commit. Fix autotest. Commit. Fix ANNIVERSARY/BDAY support in vcard 4.0. Commit. See bug #337759 Add support for importing image vcard 4. Commit. See bug #337759 Load

vcard4 photo uri. Commit. Const'ify. Commit. Add debug operator. Commit. Add LOGO vcard4 support. Commit. See bug #337759 Allow to export photo as uri. Commit. Store photo support. Commit. Add photo vcard4 support. Commit. See bug #337759 Const'ify variable. Commit. Extract code. Commit. Add photo v4 support. Commit. Add missing Q_REQUIRED_RESULT. Commit. Remove unused include. Commit. KCoreAddons Add missing include moc. Commit. Typo--. Commit. Use QCOMPARE vs QVERIFY. Commit. Update test expectation. Qt time formatting has changed. Commit. Fix the non-local license text. Commit. Switch to std::enable_if_t (API consumers now required to use >= C++17). Commit. Fix duplicate license text for known but externally linked licenses. Commit. KDBusAddons Remove unnused enum. Commit. KDeclarative Kquickcontrols: install kquickcontrolsprivate to KF_INSTALL_TARGETS_DEFAULT_ARGS. Commit. Fixes bug #510081 KDE Daemon Use newer version of session management enablement. Commit. KFileMetaData [ExtractorCoverageTest] Fix typo in clang-format processing directive. Commit. KGuiAddons Add StartupNotify=true to geo handlers. Commit. Fixes bug #510547 WaylandClipboard: Avoid overriding original UTF-8 text. Commit. Fix docs. Commit. KHolidays Use QCOMPARE vs QVERIFY. Commit. Italy: San Francesco will be a public holiday from 2026. Commit. Fixes and add days to Puerto Rico. Commit. KIconThemes [kiconcolors] Drop ActiveText. Commit. [kiconcolors] Drop Complement and Contrast. Commit. InvalidateFilter is marked deprecated in qt6.10. Commit. Use QCOMPARE vs QVERIFY. Commit. Remove version checks for Qt 6.8.0 that is now the minimum required. Commit. Avoid saving state between multiple usages of IconDialog. Commit. Fixes bug #460451 KImageformats PSD: limit memory usage on corrupted files. Commit. Fix assert on broken data. Commit. PSD: minor fixes while reading Image resource Section. Commit. IFF: fix crash on malformed files. Commit. Iff: Fix crash on malformed files. Commit. PSD: improve sections size checks. Commit. TGA: ignoring large metadata on dev area. Commit. Check device before read. Commit. Fix possible infinite loop when reading a broken jp2. Commit. On demand buffer allocation on PCHG decompression. Commit. Fix possible read overflow with malformed data. Commit. Fix wrong palette check. Commit. Use internal FP32 conversion instead of FP16. Commit. Fix read test failing on aarch64. Commit. Remove Qt version checks now that 6.8.0 is the minimum required. Commit. Fix assert on broken data. Commit. CI: Try newer libjxl for ossfuzz. Commit. Tga: Reduce Warning to Debug. Commit. Fix crash on malformed files. Commit. Fix crash on malformed files. Commit. KIO Add missing since documentation. Commit. Kfileitemactions: use actionsKey instead of name to check KAuth. Commit. [ktelnetservice] Disable session management. Commit. [kioexec] Disable session management. Commit. [kiod] Use better way to disable session management. Commit. Trashimpl: better determine trashForMountPoint for network file systems. Commit. Fixes bug #506755 KNewFileMenu: prevent emiting rejected signals when not rejected. Commit. Knewfilemenu: when in mkdir -p mode don't append a / for last dir. Commit. Kioworkers/file: check QFile::open return and early return. Commit. KCoreDirLister: Enable using mime globs. Commit. Fixes bug #450612 Reenable kdirlister test on non-Windows. Commit. KIO::MetaData::toVariant(): document internal type of returned variant. Commit. Autotests: enable connectionbackendtest. Commit. Apply 1 suggestion(s) to 1 file(s). Commit. Autotests: favicontest make a test more fault tolerant in Win. Commit. [http] Emit WorkerResult::pass() and return when a redirection is received during webdav directory listing. Commit. [http] Remember that we're doing webdav while handling redirects. Commit. Fixes bug #486790 Kifileitemactions: Remove unused QMimeDatabase instance creation. Commit. KIO::MetaData: port inline methods away from deprecated QMapIterator. Commit. Previewjob: expose standardthumbnailer fileName property. Commit. File_unix: print errnum in rename when failing. Commit. See bug #510810 Kpasswdserver: Set cancel button on dialog. Commit. JobTest: add renameFileWithNoUDSACCESS test. Commit. See bug #510567 Kfileitems: Use internal name again for sorting. Commit. Fixes bug #510470 Add typo fix commit to .git-blame-ignore-revs. Commit. Fix various typos. Commit. Use QCOMPARE vs QVERIFY. Commit. RenameDialog: wrap the question text. Commit. Ensure QNetworkReplies are always deleted with a scope guard. Commit. Fix HTTP

network error propagation. Commit. Forward all KIO error codes, not just ERR_ACCESS_DENIED. Commit. Delete network reply also when handling a redirection. Commit. Autotest/jobtest: make sure to check error return when a job fails. Commit. CopyJob: Skip permission check if there is no UDS_ACCESS entry. Commit. Fixes bug #510567 KFileWidgets: Allow saving to currentfolder with empty input field in special cases. Commit. See bug #507193 KFilePlaces: add a link to systemsettings recent config. Commit. Fixes bug #507966 [ftp] Claim that dir is writable during stat. Commit. Fixes bug #510456 KNewFileMenu: move the messageWidget below the input. Commit. Kirigami Fix documentation for AlignedSize. Commit. Add missing FrameContrast enum documentation. Commit. Fix documentation for closeDialog. Commit. Keep passive notifications inside the SafeArea. Commit. ActionTextField: don't let inline buttons accept focus on click. Commit. ActionTextField: use display property to control icons-only-ness. Commit. OverlayDrawer: Fix handle positioning on RTL layouts. Commit. Fix accessibile properties on ActionTextField properties. Commit. Support menus in bottom ActionToolbars. Commit. Feat: Expose closeDialog as a signal to the PageStack attached property. Commit. ActionTextField: adopt some good ideas from the PlasmaExtras version. Commit. ActionTextField: use standard implementation for inline action buttons. Commit. Fixes bug #484301 Use reversed open-link when needed and present. Commit. See bug #506268 PageRow: Fix dialogs not being closable via Escape. Commit. PlatformTheme: setFrameContrast value. Commit. CardsListView: Make sure it never will have horizontal scrolling. Commit. Fixes bug #477493 Cmake: Add missing icon for android. Commit. ActionsMenu: Restore the icon for submenus. Commit. PageRow: Different style for the push/pop animation/gesture. Commit. Make Separator always perfectly pixel-aligned. Commit. Fixes bug #510353 Fix up overlay sheet header content width w.r.t close button. Commit. Fix up overlay sheet's close button location. Commit. Platform: Include child Quick items when updating child PlatformTheme instances. Commit. Fixes bug #510480 Remove outdated comment. Commit. Typo fix. Commit. SelectableLabel: Only enable shortcuts if the label has focus. Commit. Platform: Skip update() in PlatformTheme if we don't have a window. Commit. Fixes bug #493921 KItemModels Remove Qt version checks now that 6.8.0 is the minimum required. Commit. KJobWidgets Keep old test code around. Commit. Disable kuiserver2jobtrackertest if HAVE_QTDBUS is false. Commit. Make the test compile + do the basic thing. Commit. Rewrite JobView code to be safer. Commit. KNewStuff Installationtest: Fix test failure on non english hosts. Commit. Don't make KNSWidgets::Button the parent of KNSWidgets::Dialog. Commit. Fixes bug #501267 Port deprecated invalidateFilter in qt6.10. Commit. KNotifications Look up notifications again after hitting application code. Commit. Fixes bug #511645 Notifybyaudio: Don't show error on CA_ERROR_DESTROYED. Commit. KPackage Use QCOMPARE vs QVERIFY. Commit. KQuickCharts Limit item count in ItemBuilder to basically uint16_t. Commit. KRunner Manager,context: remove launchcount adjustment. Commit. KService Kservice: Remove legacy X-KDE-Keywords and fix X-KDE-FormFactors. Commit. Don't warn for empty layout without merge tag. Commit. Kbuildsycoca: debug++. Commit. See bug #510287 KSVG KSvg: Add ColorScheme-Frame. Commit. KTextEditor Remove unnecessary toEdge function. Commit. Use KateViewInternal::Bias in move() signature. Commit. Use void return type. Commit. Remove the loops in the cursors. Commit. Remove unnecessary methods and const overloads. Commit. Remove operator overloading. Commit. Remove unnecessary constructors. Commit. Remove Bias::none enumerator. Commit. Make function internal to the class. Commit. Remove useless getter function. Commit. Improve the test, use higher level api. Commit. Move findMatchingFoldingMarker to KateBuffer. Commit. Cleanup includes in kateviewinternal.h. Commit. Include kateviewinternal.h less. Commit. Move include to source file. Commit. Move spellcheck function from DocumentPrivate. Commit. Remove useless function. Commit. Remove unused function. Commit. Add QJsonObject. Commit. Cleanup headers, deinline and remove unnecessary includes. Commit. Use normal function pointer, std::function is not needed here. Commit. Dont use std::pair/QPair, use named structs. Commit. Add API to obtain the editor widget from View and vice versa. Commit. Clear modeline matches if the model is irrelevant. Commit. Add new config entries to public api docs. Commit. Add

"hide-cursor-if-inactive". Commit. Add disable-current-line-highlight-if-inactive view option. Commit. Simplify kateprinter, remove unnecessary KatePrinterPrivate class. Commit. Make KateVariableExpansionManager a non QObject. Commit. Move expandText to the file where its used. Commit. Optionally return invalid cursor for coordinates outside text. Commit. Kateview_test: extract lambda coordinatesToCursor(). Commit. Explicitly convert enum to int for QString::arg. Commit. KTextTemplate Use QCOMPARE vs QVERIFY. Commit. Update test expectation. Qt time formatting has changed. Commit. KUserFeedback Properly handling large timeouts for encouragement timer. Commit. Fixes bug #511102 Use const pointer. Commit. Add missing explicit keyword. Commit. Port deprecated invalidateFilter method (qt6.10). Use beginFilterChange/endFilterChange. Commit. Use QCOMPARE vs QVERIFY. Commit. Remove old qt check. Commit. KWallet Use QCoreApplication for kwallet-query. Commit. Remove unused QtWidgets dep from API. Commit. Enable KCrash for kwalletd6. Commit. KSecretd: Use modern way of disabling session management. Commit. Kwalletd: Disable session management. Commit. Fix dismissing prompt for Unlock. Commit. Use securelySeeded like in KWalletPortalSecrets::generateSecret. Commit. Backend: Simplify random gathering to QRandomGenerator. Commit. KWindowSystem Platforms/wayland: Prevent reinstalling window effects with the same parameters again. Commit. Kwaylandextrastest: Port to UI file. Commit. Remove Qt version checks now that 6.8.0 is the minimum required. Commit. Fix creating empty future. Commit. Fix KWaylandExtras::xdgActivationToken when KWindowSystemPrivateV3 isn't available. Commit. KXMLGUI Use CamelCase Qt include. Commit. Use QCOMPARE vs QVERIFY. Commit. Switchlanguage: remove duplicates from languageList. Commit. Network Manager Qt Add missing documentation comments for WifiP2P. Commit. Fix since documentation for WifiP2P. Commit. Fix enum value for WifiP2P. Commit. Deprecate unused, Java-style iterator NMStringMapIterator typedef. Commit. Add WifiP2P device. Commit. Implements feature #502159 Prison Videoscanner: Remove Qt5 note. Commit. QQC2 Desktop Style PlasmaDesktopTheme: Set frameContrast on change. Commit. Harmonize delegate tooltip code. Commit. SpinBox: Make paddings integer to avoid potential polish loops. Commit. Fixes bug #510758 Remove Qt version checks now that 6.8.0 is the minimum required. Commit. Solid Udisks: Port to DBus Object Manager. Commit. Sonnet Remove double margins around groupbox. Commit. Fix load default ignore list when we call slotDefault method. Commit. Add missing [[nodiscard]]. Commit. Use [[nodiscard]]. Commit. Show headers in qtc. Commit. Ensure parent is alive when removing eventfilters. Commit. Fixes bug #492444 Add SpellCheckDecorator destruction autotests. Commit. Fixes bug #492444 Syndication Show headers in qtc. Commit. Syntax Highlighting QFace: Change type highlighting and add fully qualified cases. Commit. Implement D2 syntax highlighting. Commit. Cpp: Update classes for Qt 6.10. Commit. Add Starlark syntax highlighting. Commit. Add new features to QFace highlighting. Commit. Threadweaver Use QCOMPARE vs QVERIFY. Commit.

- Convert from Qt to Qt for MCU – Qt AI Assistant 0.9.7 released (2025/11/13 10:33)
  Stretching your Qt UI to devices powered by microcontrollers (MCU) is now even easier! The Qt Quick Ultralite Converter of the Qt AI Assistant translates a QML to Ultralite-compliant code. It brings core Qt graphical features to resource-constrained embedded systems, streamlining application development and deployment.
- Akademy 2026 - Celebrating KDE’s 30th Anniversary in Graz (2025/11/13 09:21)
  Akademy 2026 will be a special edition, marking the 30th anniversary of KDE! This milestone event will take place at the Graz University of Technology in Graz, Austria. This birthday edition of Akademy will continue to bring together contributors, users, partners, and friends of KDE to reflect on three decades of collaboration, innovation, community growth, and commitment to Free Software. Just like previous years, Akademy 2026 will be a hybrid event, offering both on-site and online participation. We will be announcing the exact dates soon. Until then, follow us on

Mastodon and Lemmy for the latest Akademy updates! About Graz Graz is the second-largest city in Austria and the capital of the federal state of Styria. Known for its well-preserved old town, which is a UNESCO World Heritage Site, Graz offers a blend of historic charm and modern vibrancy. The city is home to numerous cultural attractions, including the iconic Schlossberg with its clock tower, the Kunsthaus Graz, and the Murinsel, a unique architectural feature on the River Mur. Graz is also renowned for its educational institutions, particularly the University of Graz and the Graz University of Technology, making it a hub for innovation and research. The city's lively atmosphere, beautiful parks, and rich culinary scene make it an ideal destination for both leisure and professional visits. About Akademy For most of the year, KDE, one of the largest free and open software communities in the world, works online communicating over email, instant messaging, video-conferencing, forums and mailing lists. Akademy provides all KDE contributors with the opportunity to meet in person to foster social bonds, work on concrete technology issues, discuss new ideas, and reinforce the innovative, dynamic culture of KDE. Akademy brings together artists, designers, developers, translators, users, writers, sponsors and many other types of KDE contributors to celebrate the achievements of the past year and help determine the direction for the next year. Hands-on sessions offer the opportunity for intense work, bringing those plans to reality. The KDE community also welcomes companies building on KDE technology to Akademy, as well as those who are looking for opportunities.

- What’s next for Aurorae? (2025/11/13 07:42)
  Aurorae is a decoration engine that allows you easily using third party decoration themes from KDE Store. Aurorae decoration themes at KDE Store (although there are some decoration themes that don’t use Aurorae, e.g. Klassy, but a good chunk of themes found in the Plasma 6 Window Decorations category still use it) It’s been around for quite a while and it has a plenty long history. However, in the recent years, it has been somewhat neglected. The UI trends changed, e.g. rounded corners are all the rage now, but there has been no changes in Aurorae to allow theme creators to follow those trends more easily. There are also performance issues. In comparison to the default Breeze decoration theme, it performs quite poorly, unfortunately. What is Aurorae anyway? In Plasma, we have a C++ library that’s used to implement window decorations called KDecoration. Both Breeze and Aurorae use KDecoration, but the main difference between the two is that the former directly implements a window decoration that follows Breeze style, while the latter is just a very themeable window decoration. Aurorae supports both QML and SVG themes. With a QML theme, you need to write some QML code to define how the window decoration should look and behave. With an SVG theme, you need to provide a bunch of SVG files that specify how the window frame and various buttons look. Under the hood, SVG themes are effectively built as QML themes. QML is pretty cool because with a few lines of code, you can get something that works and looks very decent. But for our usecase, it’s also a heavy tool, and due to the way how QtQuick works, it’s very challenging to have proper fractional scaling support. To be fair, normal applications that use QtQuick are mostly fine, it’s just that we have a pretty unique usecase where we need full control where every individual pixel gets painted. Aurorae V2 Recently, I started a rewrite of Aurorae with a few goals in mind. The first goal is to improve performance so KWin doesn’t struggle when you resize a window (just to be clear, no, this doesn’t mean that there are performance issues in QtQuick. The way QML decorations are rendered is inefficient. We render a QtQuick scene in an offscreen texture, then download its contents, and then upload its contents in another texture. These roundtrips kill performance, and they are necessary because of KDecoration API constraints). The second goal is to open up the road for fractional scaling support improvements, like fixing misaligned window border edges or gaps between the decoration and the window contents. The third goal is to prepare the foundation for future improvements and to provide more tools so artists can create themes that reflect “modern” stylistic tendencies. In this rewrite, raw KDecoration and KSvg APIs are used. This makes aurorae decorations quite lightweight and it significantly improves performance. Some fractional scaling issues have already been fixed, while

others still need more work. Window decorations are rendered on the CPU side. While, yes, it will be nice to have everything done on the GPU side, doing things on the CPU side is also not that of a big bottleneck right now. In the future, we may follow up on doing more things on the GPU, but for now, it is not high priority. The main focus has been on improving performance issues and preserving compatibility with the previous implementation of SVG decoration themes so decorations look more or less the same way. There may be some (unintentional) differences, but they should be very minimal. What about QML decoration themes then? The main idea behind them is brilliant, but after so many years, there are not that many window decoration themes that use QML. In either case, the original (V1) and rewritten (V2) aurorae engines live side-by-side, but the future of V1 is unclear. QML and SVG decoration themes will use V1 and V2, respectively. Future plans At the moment, the goal is to continue polishing and optimizing V2. There are certain limits how far we can push things due to the theme format. Some parts of the theme force us to do some things, which ideally we shouldn’t do. It’s highly likely (not saying this for sure though!) that there will be a V3, which is going to address some theme limitations and add support for new features, for example rounding bottom window corners or outlines. If you’re a decoration theme creator and would like to see some particular feature in Aurorae, feel free to reach out to me (@zzag) on Matrix (e.g. in #kwin) or file a feature request at https://bugs.kde.org (product: kwin; component: aurorae).

- Qt Champions 2025 (2025/11/13 07:30)
  Dear Qt community! with seven weeks to go until an exciting year 2025 will have passed, we are pleased to open the nomination period for the Qt Champion 2025 award!See this Wiki page for more information about the award and its nomination process: https://wiki.qt.io/Qt_Champions_2025We're looking very much forward to reading about your favorite candidates! CheersAxel
- Value Types: Reputation and Influence (2025/11/11 16:11)
  This is the forth article of the series describing the open source value flow model. We'll focus on the value types: Reputation and Influence. We'll cover how to measure and report on them.
- Plasma Setup Security Improvements (2025/11/11 00:00)
  Plasma's first-run experience (FRE) / out-of-box experience (OOBE) has seen significant improvements in security recently. Although first off I think I maybe hadn't mentioned yet how the project was renamed. Previously known as "KDE Initial System Setup" (KISS), the project has been rebranded to "Plasma Setup" and now sits nicely alongside other system projects like "Plasma Desktop", "Plasma Mobile", "Plasma Keyboard", etc. Enhancing Security in Plasma Setup We received a notice of potential security issues from the folks at openSUSE, which have now been addressed. This sort of thing is a great example of why it can be so difficult to provide ETAs and timelines for software development: unexpected issues often arise that need to be addressed before other planned work can proceed, and these issues can take time to investigate and fix properly; in this case, the security issues required careful review and testing to ensure that they were resolved without introducing new problems, and delayed our initial release by weeks. I had very little experience with this sort of security-minded defensive programming before this, so it was a great learning experience for me personally as well. It required a whole lot of reading and research to understand the best practices and principles involved, and I definitely have a better feeling for how to think about defensive programming in the future. It amazes me the kinds of things people will try to do to break software, and many of them (like path traversal attacks) are things I would never have thought of on my own! I'd like to thank the openSUSE security team for responsibly disclosing these issues, and for their patience while we worked through them. Their help has made Plasma Setup more secure for all users, and I appreciate their dedication to improving the security of open source software. A massive thank you specifically to Matthias Gerstner for the multiple rounds of detailed and thoughtful reviews and suggestions

on the MR to address these issues. Your help was invaluable, and it was a pleasure working with you! □♀ □ □ Looking Ahead Plasma Setup is nearly ready for initial testing and adoption. There are a couple more items to wrap up, but (barring further unforeseen delays!) we are very close to being able to release it for early adopters to try out! □

- Krita Monthly Update - Edition 32 (2025/11/11 00:00)
  Welcome to the October 2025 development and community update. Development Report Text Rework Progress The Text Tool's Tool Options have been overhauled. There's now a button to access the Text Properties docker, along with an option to create new texts with current properties or with a style preset. Options to switch between using visual or logical cursor direction for bidirectional text, and pasting rich or plain text have also been added. (Change) The other addition is Type Setting Mode, which shows transform handles for the font size and baseline-shift. With preformatted or pre-positioned text, character transforms are also possible. Holding Shift shows different baselines to switch to. (Change) In the area of file formats, basic support has been added for PSD text layers, vector masks, vector strokes, vector parametric shapes, and guides. (Change) Wolthera discusses these changes in the Text Tool thread and asks again for feedback on a proposal for splitting character and paragraph properties. Touch Input Fixes Carsten has continued to fix issues with touch input in the Stable branch. Long-press handled has been further improved. Long-pressing a slider-spinbox no longer shows text selection handles when not in text edit mode. The long-press distance is now calculated correctly, so a slight movement won't cancel it. The kinetic scrolling timeout no longer adds onto the long-press timeout. (Change) Kinetic scrolling by left-click has been disabled on the animation timeline to not interfere with dragging frames and other operations. (CCbug report) (Change) Popup-at-cursor widgets such as the Selection Action Menu now appear at the touch location instead of cursor location. (Change) The Edit Shapes Tool now works properly with touch, instead of only making selections. (bug report) (Change) Wayland Support Basic HDR support for the canvas on Wayland has been implemented by Dmitry. Testing instructions and discussion of issues can be found in the forum thread. (Change) Plans for 5.3.0's Upcoming Release Krita 5.3.0 is scheduled to enter feature freeze on November 21st. This means no new features will be accepted for the next version, and developer focus will shift to finishing features already in progress and fixing bugs. After a bugfixing period of a few months, the first beta testing release is currently planned for February. Community Report October 2025 Monthly Art Challenge Results 14 forum members took on the challenge of the "The Burden of Power" theme. And the winner is... The Burden of Power by @Famouzy Be sure to check out the second piece as well! The October Art Challenge is Open Now For this month's theme, winner @Famouzy has chosen "Civilization Engulfed by Nature". Featured Artwork Best of Krita-Artists - August/September 2025 This month's Best of Krita-Artists Nominations thread received 21 nominations of forum members' artwork. When the poll closed, these five wonderful works made their way onto the Krita-Artists featured artwork banner: Autumnal Street by @Paulo Young Escherstein! by @jimplex Bee Macro by @Brian_Bigelow Interstellar Gura by @RavioliMavioli Peaceful Stream by @CrazyCatbird Best of Krita-Artists - October/November 2025 Take a look at the nominations for next month, and suggest your favorite latest artworks to be featured. Don't forget to vote when the poll opens on November 11th! Ways to Help Krita Krita is Free and Open Source Software developed by an international team of sponsored developers and volunteer contributors. That means anyone can help make Krita better! Support Krita financially by making a one-time or monthly monetary donation. Or donate your time and Get Involved with testing, development, translation, documentation, and more. Last but not least, you can spread the word! Share your Krita artworks, resources, and tips with others, and show the world what Krita can do. Other Notable Changes Other notable changes in Krita's development builds from October 20, 2025 - November 11, 2025. Stable branch (5.2.14-prealpha): Android: Make app fullscreen by default. (Change, by Carsten Hartenfels) Canvas Input Shortcuts: Add Toggle Eraser Preset to canvas input shortcuts. (Change, by Carsten Hartenfels)

Unstable branch (5.3.0-prealpha): Blending Modes: Add Marker blending mode. When used on a brush in Build up painting mode, it increases the layer's opacity only when the stroke's opacity is greater while mixing the colors. It's similar to Alpha Darken, but adheres to alpha lock/inherit alpha and interpolates between colors cleanly. (Change, by Carsten Hartenfels) Shortcuts/Toolbars: Add actions for each Transform Tool mode. (Change, by Stuffins) Toolbars: Toolbar actions' icons can now be custom-picked in Configure Toolbars, useful for actions that do not have icons by default. (Change, by Pavel shlop) macOS: Sign and notarize nightly builds, allowing them to be run without workarounds. (Change, by Ivan Yossi) Nightly Builds Pre-release versions of Krita are built every day for testing new changes. Get the latest bugfixes in Stable "Krita Plus" (5.2.14-prealpha): Linux - Windows - macOS (unsigned) - Android arm64-v8a - Android arm32-v7a - Android x86_64 Or test out the latest Experimental features in "Krita Next" (5.3.0-prealpha). Feedback and bug reports are appreciated!: Linux - Windows - macOS - Android arm64-v8a - Android arm32-v7a - Android x86_64

- KDE Plasma 6.4.6, Bugfix Release for November (2025/11/11 00:00)
  Tuesday, 11 November 2025. Today KDE releases a bugfix update to KDE Plasma 6, versioned 6.4.6. Plasma 6.4 was released in June 2025 with many feature refinements and new modules to complete the desktop experience. This release adds two months’ worth of new translations and fixes from KDE’s contributors. The bugfixes are typically small but important and include: View full changelog
- Mnietballs, again (2025/11/10 23:00)
  Some time back I posted a bran muffin recipe. That recipe uses eggs, and I received a comment by email, perhaps from Gregor S. They pointed out some things about the egg-milk-industrial complex and how I should avoid using eggs. That annoyed me a little, although it was written respectfully and factually. Anyway, I’ve given it some thought and experimentation, and here’s a vegan recipe for mnietballs (vaguely Dutch for “not meatballs”). I wrote a mnietballs recipe in 2023, which uses eggs as well – so that recipe is ovo-lacto-vegetarian, while this one is vegan. Vegan, but now the soy-industrial complex is involved. 1 tea cup (I have a specific glass one in mind, probably 220ml) of dried soy chunks. Soak them in 1 cup of warm water for 10 minutes, then squeeze out all the water. 1 tablespoon of chia seeds and 1 tablespoon of whole linseed, with 2 tablespoons of water. Let stand for 10 minutes until it all gloms together. 1 tea cup of rolled oats. whole wheat flour as needed. 2 tablespoons of ketjap manis. spices to suit (I like the tandoori masala I get at Toko Weuro). optional, finely chopped onion. optional, dried chopped parsley. Mix it all together and knead it to a lumpy paste. Add wheat flour until it is fairly stiff and you can shape it into balls about 3cm in diameter. Let stand for a while for it to glom together better. Optionally roll each ball in some breadcrumbs. Fry in plenty of oil.
- Alternative Timezone Naming in Calamares (2025/11/10 23:00)
  Calamares is a Linux system installer. During installation, it asks the user where they are on the globe, in order to set the timezone correctly on the installed system. Calamares displays the nearest timezone after you click on a map. I would like to leverage that a little for social good (or at least a tiny bit of awareness). The last time I wrote about the Calamares timezone selector, I also said it is terrible. One thing I do like about the timezone selector is that it supports translating the name of a timezone. That way, even though the string in the timezone database is Europe/Kiev – a relic of the time-period that the timezone database was conceived – it displays the correct Europe/Kyiv. That’s when you run Calamares in English, anyway. I added timezone translations to Calamares because a friend asked for it, and then did a couple of Dutch translations. That is because Dutch has exonyms (names in Dutch for other places) and calls Berlin, Berlijn and Paris, Parijs. There is a timezone Europa/Berlijn. The official Dutch name – last I checked – of Kyiv is Kiev, though. The current translations addresses only a tiny fraction of the timezones, most of them ones that I have personally paid attention to, and I’d like to change that. So here is the social project: I want, for every

timezone-location (in FreeBSD, those are listed in /usr/share/zoneinfo/zone.tab) a local name. Please send them to me by email or as a change request in Calamares on Codeberg, the relevant file is tz_local.ts. I imagine a lot of the ones in Europe will be unchanged, although I feel like Europe/Chisinau is missing accents, Europe/Athens should be Αθήνα. Those are relevant, but I'm more interested in imported names. The exonyms (what others call a place, like "Berlijn") could be endonyms (what is the local name, like "Berlin"). For everywhere where the name of the timezone was imported, I want to know what it would be locally, written locally. From Africa/Harare (this one might already be accurate, but for comparison purposes, pretend it was Africa/Salisbury, like in an atlas from 1922) to Australia/Broken_Hill to America/Edmonton (which could plausibly be ᐊᒥᐢᑿᒌᐚᐢᑲᐦᐃᑲᐣ if we pick the Cree name for the area) to America/Santiago to Asia/Dubai (دبَيّ in Emirati Arabic) they all deserve a local name and I'd like to make that possible in Calamares.

- This Week in KDE Apps (2025/11/10 08:21)
  Quality of live improvements in Kate, basic HDR support in Krita on Wayland and touch improvements in PhotosWelcome to a new issue of "This Week in KDE Apps"! Every week (or so), we cover as much as possible of what's happening in the world of KDE apps. As part of our yearly fundraiser, you can adopt one of KDE's apps and we can share with the whole world how awesome you are and how much you're doing to support us. Getting back to all that's new in the KDE app scene, let's dig in! Multimedia/Graphics Applications Photos Image Gallery Anders Lund added support for navigating between images when zoomed in using a touchscreen or stylus in Photos (25.12.0 - link). He also made the viewer auto-zoom when releasing a pinch when appropriate (e.g., the image is now smaller than the view) (25.12.0 - link). Anders also enabled the slideshow feature on mobile (25.12.0 - link). Travel Applications KDE Itinerary Digital travel assistant Jonah Brüchert implemented auto-resolving from railway station names to coordinates based on Nominatim (the geocoding engine powering OSM) (25.12.0 - link). Volker Krause added support to the extractor for citycity.se, Comboios de Portugal, and Wiener Linien barcodes, and improved a bunch of existing extractors (25.12.0 - link 1, link 2, ...). PIM Applications Akonadi Background service for KDE PIM apps Christoph Erhardt fixed a segmentation fault when migrating an existing Akonadi database to SQLite (25.12.0 - link). Nicolas Fella dropped the barely used feature to show an "About Data" dialog in Akonadi resources (25.12.0 - link). Office Applications Okular View and annotate documents Jack Barmes added new bookmark actions to add and remove books in various context menus in Okular (26.04.0 - link). Utilities Applications Kate Advanced text editor Waqar Ahmed improved Git support in Kate. The list of branches now shows the latest activity (25.12.0 - link). Additionally, he improved the Quick Open dialog, and it is now possible to jump to a specific line and column by entering something like 10:5 (25.12.0 - link). Dennis Lübke added a plugin to transparently edit encrypted text files with GPG in Kate (25.12.0 - link). Social Applications NeoChat Chat on Matrix Joshua Goins added an informational Keyboard Shortcuts settings page to NeoChat (25.12.0 - link). Joshua also improved the design of the room notification settings and made it more consistent in terms of wording with the context menu used to configure notifications (25.12.0 - link). Among a multitude of other small fixes, he improved the user experience related to the basic Jitsi meeting button to show whether a meeting is in progress, and to only enable it when the user has permission to start a meeting (25.12.0 - link). Creative Applications Krita Digital Painting, Creative Freedom Carsten Hartenfels added a way to trigger the "toggle eraser preset" action via touch gestures in Krita (Krita 5.2.x - link). Wolthera van Hövell ported the text tool dock to QML (link). Dmitry Kazakov implemented basic HDR support on Wayland (link). Games Chessament Chess tournament manager Manuel Alcaraz replaced the hamburger menu with a menu bar (link). Third-Party Applications Easy Effects - Audio Effects for PipeWire Applications Easy Effects 8 was released and now uses Kirigami! Read the full announcement for details. …And Everything Else This blog only covers the tip of the iceberg! If you're hungry for more, check out Nate's blog about Plasma and be sure not to miss his This Week in Plasma series, where every Saturday he

covers all the work being put into KDE's Plasma desktop environment. For a complete overview of what's going on, visit KDE's Planet, where you can find all KDE news unfiltered directly from our contributors. Get Involved The KDE organization has become important in the world, and your time and contributions have helped us get there. As we grow, we're going to need your support for KDE to become sustainable. You can help KDE by becoming an active community member and getting involved. 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. There are many things you can do: you can help hunt and confirm bugs, even maybe solve them; contribute designs for wallpapers, web pages, icons and app interfaces; translate messages and menu items into your own language; promote KDE in your local community; and a ton more things. You can also help us by donating. Any monetary contribution, however small, will help us cover operational costs, salaries, travel expenses for contributors and in general just keep KDE bringing Free Software to the world. To get your application mentioned here, please ping us in invent or in Matrix.

- Plasma Mobile 6.5 (2025/11/10 00:00)
  A mobile-centric look at the Plasma 6.5 release About three weeks ago we released Plasma 6.5 and it's high time we talk about the plethora of improvements and bug fixes that arrived in Plasma Mobile and related projects. Let's not delay any further and get right into the juicy details! Waydroid Integration While our end goal is obviously KDE for world domination and a resulting breadth of native apps, we're not quite there yet and until then we wanted to make it easier to use and integrate apps running through Waydroid into your Plasma Mobile system. To that end, Florian made it so you can now set up and manage your Waydroid install right from the comfort of your settings app and turn on/off the Waydroid container from the quicksettings dropdown. (Florian Richer, Link 1, 2, 3, 4, 5, 6, 7) Lockscreen The lockscreen is one of the most often used and seen parts of a phone's UI and it has two jobs: Securely lock your device, but ultimately, get out of your way. So far we succeeded at the first of these jobs, but in 6.5 Devin made sure we do in the second as well. Plasma Mobile's lockscreen will now load much faster, due to reusing the existing status bar and action drawer process from the shell, instead of loading them separately each time the lockscreen is loaded. This also has the positive side effect of notifications now properly being synced between the lockscreen and unlocked shell. (Devin Lin, Link) Florian also worked on some other nifty additions: Add double tap to lock Polish UI feedback for lockscreen actions (Florian Richer, Link 1, 2, 3) Folio Homescreen Micah improved and expanded the background blur effect for both the Folio and Halcyon homescreens, resulting in more consistent use of blur. (Micah Stanley, Link) Devin put in some work to make Folio more keyboard navigation friendly, as well as improving it in various places under the hood: from cleanup to performance improvements there's a good mix of changes here (Devin Lin, Link 1, 2, 3, 4, 5, 6) Meanwhile, Florian added double tap to lock to Folio, as well as a number of other improvements. (Florian Richer, Link 1, 2, 3, 4, 5) Halcyon Homescreen Devin unified some code paths between Halcyon and Folio, and added a small settings page to Halcyon. Like with Folio, he also ported Halcyon to use a new applet registration method which allows it to be precompiled by qmlcachegen (Devin Lin, Link 1, 2, 3) Settings There's a good bunch of stuff here as well: Separated out the navigation settings from the Shell settings page to its own page and added a small tutorial for the gesture navigation mode there. (Luis Büchi, Link) Added a setting to change the maximum number of quicksetting columns, really useful for tablets! (Sebastian Kügler, Link) Devin made sure mobile settings pages only show up on form factors that they make sense on so we stop showing the mobile shell settings on desktop. (Devin Lin, Link 1, 2) Taskswitcher Luis has worked on the taskswitcher internals to make it more compatible with qmlcachegen. This lays the groundwork for improving its performance while also making the code more maintainable for the future. (Luis Büchi, Link 1, 2, 3, 4) Haptics Devin has done the first step on porting over our haptics plugin to use feedbackd as backend. In the future, this will allow us more fine-grained control over haptic feedback, as well as better cross-desktop compatibility (Devin, Link) Action

Drawer Besides Devin's work on making the action drawer overlay over the lockscreen mentioned above, there's also been some more work on improving performance of it and its contained quick actions. For all of you running multi-monitor setups with Plasma Mobile and/or regularly docking and undocking, thanks to Sebastian there's now a quick settings to configure multi monitor layouts when several monitors are connected. (Devin Lin, Link) (Florian Richer, Link) (Sebastian Kügler, Link) Envmanager The envmanager, responsible for keeping track of the settings environment of the shell has gotten more robust while improving how Plasma Mobile and Plasma Desktop coexist: there is now less cross-talk between potentially conflicting settings between the two shells making a hybrid setup (like on a 2-in-1 laptop/tablet combo) easier to use. In light of the work on Plasma keyboard, envmanager now also properly supports changing of the selected virtual keyboard for the mobile session. (Devin Lin, Link 1, 2, 3, 4, 5, 6) UI Polish We've also spent some time improving the general look and feel of the UI. This includes updating some elements to use more appropriate theme colors, using system-wide animation durations and better layouting to reduce overlapping text/UI controls. (Devin Lin, Link 1, 2, 3, 4) (Micah Stanley, Link 1, 2, 3) (Florian Richer, Link 1, 2) (Luis Büchi, Link) Plasma Keyboard While not technically part of the 6.5 release cycle, we've recently also released the first unstable version (0.1.0) of our new virtual keyboard. While it's not ready for prime-time just yet, progress is quick and it's already something enthusiasts may want to tinker with - so tinker away and do let us know of any feedback you might have! ...and there’s much more. To see the full list of changes, check out the complete changelog for Plasma 6.5. Contributing Do you want to help with the development of Plasma Mobile? We are a group of volunteers doing this in our free time, and are desperately looking for new contributors, beginners are always welcome! See our community page to get in touch! We are always happy to get more helping hands, no matter what you want to do, but we especially need support in these areas: QA Testing Telephony (Calling and SMS) Camera App development (Photo Viewer, Browser, Audio Recorder, Games, etc.) Shell work You can also check out our Plasma Mobile issue tracker for more details. Even if you do not have a compatible phone or tablet, you can also help us out with application development, as you can easily do that from a desktop! Take Plasma Mobile for a spin! Check out the device support for each distribution and find the version which will work for your device. If you have any further questions, view our documentation, and consider joining our Matrix channel. Let us know what you would like to work on or where you need support to get going! Our issue tracker documentation also gives information on how and where to report issues.