

# X WINDOWS, WAYLAND AND THE FUTURE OF LINUX

Will 2014 be the year of Wayland on the Desktop? Daniel Stone thinks so, and **Ben Everard** caught up with him to find out why.

**X** Windows first came out 30 years ago, and it's been at the centre of Unix and Linux graphics ever since. However, times have changed, and X often struggles to take advantage of the capabilities of modern graphics hardware.

Linux needs a new display server, and Wayland is the software that many people hope will keep Linux graphics moving forward.

Linux Voice headed over to Cambridge to meet Daniel Stone – a long-term X contributor, prominent Wayland developer and graphics lead

at Collabora. He gave us the low-down on what the problems are with X, why Wayland is better, what's happening with the project and when we're likely to see it come to fruition.

Along with helping regular Linux users get snappier graphics, Collabora are working with the Raspberry Pi foundation to use Wayland to unlock the graphical powers of the little computer and create a much smoother desktop environment. Despite a lot of talk, it hasn't yet shipped in a major distro, though this should change by the end of the year.



**LV** What will a regular Linux user notice when they install a distribution with Wayland for the first time?

**Daniel Stone:** Our main aim is to get out of the way! With X, you've got the X server, the window manager and the clients. You've kind of got a window manager and a compositing manager and there's a three-way disaster.

We've tried to make that whole architecture as simple as possible – or make the protocol as simple and straightforward as possible. I guess that's the big thing: always being smooth and responsive and good.

**LV** Do you have any idea when we're likely to start seeing Wayland... Other than on Rebecca Black OS of course.

**DS:** Oh my God, why is it called that? It's starting to appear. Fedora 21 – it'll be

there completely solid. I'm running Gnome Shell with Wayland on my laptop, and it's actually really quite good. I hadn't really tracked any of Mutter or Gnome shell, so I was quite surprised by how quickly they were able to get it up.

**LV** What have you found the biggest challenges to be in developing Wayland?

**DS:** It's sort of been this 10-year ongoing thing. It's the culmination of everything we've done in X since about 2004. It's not called X12, because it's a very different windows system [from X11], but a lot of the work we did to look at things like kernel mode setting, DRI2 – all of that kind of thing – they're all really designed to get X out of the way as much as possible. Partly because architecturally it was the right thing to do, but then also there's an explicit goal to make it possible to actually develop

alternate windows systems. If you go back to kind of Berlin, Fresco or KGI, they just suffered really badly because there were no video drivers at all. There was barely any toolkit support. Definitely no 3D support. So yeah, just getting over that mountain was really difficult. But now half of the Mesa internals have been rewritten. It's just been a really really long slog.

Just trying to convince people that it's actually ready and usable has been really tough as well. Around about a year ago we had this huge spurt in development – to be fair I think you can partially attribute that to Mir, but a lot of that was that Red Hat had finished an enterprise cycle and suddenly they had a lot of guys come out and say "now we're actually free to start working on this". That really got us from a stage of "we've got a windows system that looks like it should work and we've got these



“So far Nvidia are the only ones who’ve tipped their hat and said “yes, we are working on Wayland support.”

toy clients; GTK kind of works and we’ve got this toy desktop environment” to “we know this actually works and runs well on a proper fully-fledged desktop” – that’s only quite recent.

**LV Intel has been quite active. How have the other hardware manufacturers responded to Wayland?**

**DS:** For 2D support, if you have kernel mode setting [KMS], then that’ll just work fine. For 3D support, Nvidia have said publicly that they’re working on Wayland. Then you’ve got Arm with the Mali GPU, Imagination, PowerVR and

Avanti, Qualcomm and Broadcom – if you want 3D, all of those need to add support to their drivers. That’s about as much as I can say, unfortunately.

**LV Is it likely to be a difficult job to get drivers to run Wayland on the desktop?**

**DS:** It shouldn’t be! If it isn’t seamless then we’ve done something really badly wrong. On the desktop it should be totally fine. Mesa supports it fine, and they all have KMS support, so in that sense, it should be completely seamless. On mobile, it’s just up to the manufacturer. So far, Nvidia are the only ones who have actually tipped their hat and said ‘yes we are working on Wayland support’.

**LV When you say on mobile, which mobile platform are you talking about?**

**DS:** Mobile is a lazy shortcut for Arm-based! Wayland has been shipped in a set-top box a couple of years ago, a TV and also the Jolla phone, so in terms of actual mobile, we’re in a really good spot because we worked together with the Jolla guys to do libhybris [a compatibility layer to allow glibc-based software to run on bionic libc (the Android C library) systems]. We sort of did two halves of it, then they glued it together where you can reuse the binary blob – or if you have source, even better. You can re-use the Android vendor drivers you get, and that just adds Wayland to any Android-capable platform. That’s used on the Jolla Sailfish phone.

**LV Do you mind telling us a bit about how you got started contributing to open source and Free Software?**

“If the switch to Wayland isn’t seamless then we’ve done something really badly wrong.”

Wayland has brought huge improvements to the graphics of low-power devices such as the Raspberry Pi.



**DS:** I think I got started with either KDE or Debian, I can't remember how.

**LV Was it KDE on Debian?**

**DS:** There was also that, but that was later on. Yeah, I just trundled along doing a couple of small KDE bits on their IM client (Kopete), and a lot of packaging in Debian. For some reason, I ended up maintaining Apache 2 and KDE and then started working on XFree86, which I kind of got tricked into.

**LV So you got all the good jobs!**

**DS:** Yeah! I was working at a college at Melbourne Uni and they said "We've just got this new lab and we need a new version of X to support the new Intel GPU. Do you reckon you could do that?" I thought that'd take about a week, then a year later... That was right about the time of the XFree86/X.org split. Somehow I did the modular build system for the X server, then I did a bunch around that, and I gradually got dragged into working on the core server. It was pretty much 11 years ago... I'm still here! It was a miss-spent youth.

**LV How much easier is Wayland going to be to package than XFree86 was?**

**DS:** So much easier! It was a disaster

when we had the old build system with everything whacked in one and it took a day to build. Modularisation was really great, but we kind of took it one step too far and ended up with something like 380 different modules. Most of them never change, but it went a bit too far in the other direction. Now we've just got Wayland and Weston and whatever GL stack you have. Fingers crossed, it won't get out of hand!

**LV From a user's perspective, will they need to know the compositing manager from the window manager?**

**DS:** We merged those! The key architectural change is that the window manager and the compositing manager got merged into the display server, so we've got Weston, which is our reference compositor, and that has plugins. It's got a super basic UI on top of that. The IVI [In-Vehicle Infotainment] guys have been working on their own shell for automotive stuff.

The new desktop we released for the Pi, called Maynard, is a Weston module. It runs in-process, which is really nice, but by the same token, we don't have to re-invent the compositor.

Other projects like Kwin – I think – are doing their own compositor. Definitely

Gnome Shell and Mutter are separate. It shouldn't look any different than it does now where you just pick a window manager basically.

**LV What would you say you've learned from XFree86? It's one of the best pieces of software, and it's adapted and interconnected everything. Is there anything from the success of X that's gone into Wayland? Like that the client and the server are the wrong way around?**

**DS:** We kept that one! I guess it's easiest to look at the things we did differently. X has this awkward thing where the clients mostly render everything, but then you get the server to do about a third of it.

Longevity is kind of hard to pick because, as I said, we spent 10 years digging all of these bits out of X, and before that, there was no real alternative either. It's partially artificial. I guess it taught us that no matter how high you make the barrier to entry, people will still do it. I'll have to think about it...

**LV We suppose X always did the job, despite being complicated.**

**DS:** It's definitely mostly good enough. It was quite easy to push it in some quite unexpected directions, so it was this

infinitely flexible framework that you could make do almost anything.

**LV** We remember being amazed that you could get anti-aliased fonts. Just the fact that you could patch something old to come up with something new. Then wobbly windows as well!

**DS:** That was a blessing and a curse! Inside the X server, it's not a clear, straightforward process that everything flows through. You've got this one

**“We just kept adding more layers... in the end it made it really hard to work with.”**

massive structure that everything hangs off which has about 50 function pointers in it, and everything goes through those. We just kept adding more and more layers. It was great that it allowed us to do that, but in the end it made it really hard to work with.

Things like composite were done with the core being completely unaware that the composite was ever there. We just wrapped everything up and did a couple of tricks on the way in and undid them on the way back out. It made it incredibly fragile and hard to work with.

Things like minor rendering bugs (like when the screen flashes black for a second) – you wonder why that happened and three weeks later you come out with the answer.



**“Intel are hugely involved in Wayland development – they are completely behind it.”**

**LV** Will there be an easy equivalent to VNC?

**DS:** There's already an RDP [remote desktop protocol] back-end for Weston, and the RealVNC guys have been on the Wayland list looking into doing something quite similar. One of the things we did – it's almost a happy accident, but partly by design – makes it much easier for the compositor to stream out windows externally, so you can do things like the hangouts or Skype-style screen show.

**LV** How much work is it to convert an X driver to Wayland?

**DS:** Not a lot if you already have something that can already to a composited X server, which is fewer than you'd think.

**LV** Is it just renaming a few of the calls?

**DS:** Simplifying a lot of them! One of the things we did which ends up making it a lot easier for driver manufacturers is – you know previously the X server would load the graphics driver which was specific to the 3D hardware and the display controller, and then you had your client-side GL library and they had to agree on a protocol. You had to get this marriage between what you had on the client and the X server driver you had, and your display controller as well. We broke that out quite nicely. That was something that took a couple of iterations of refining before we got right, but it's now split out so that all of the Wayland 3D stuff lives in the GL library rather than the X server that had an API that changed about every six months.

**LV** Will Wayland make it easier for Nouveau, Lima and the open driver projects?

**DS:** Most of their complexity is unrelated to be honest. We've got a nice clean split between the windowing system and the actual rendering – which is reflecting in the GL and GLES – so it doesn't make any odds to them to be honest! All that infrastructure's in core Mesa and they don't have to deal with it – much as I'd love to say yes.

**LV** What are your thoughts on Mir?

**DS:** They're doing their own thing. That builds on ten years of us extracting stuff

out of X, and also libhybris which we originally did to run Wayland on Android – they were able to take that and run Mir on top of it instead. We're both standing on the same foundations.

**LV** Do they re-use your driver hooks?

**DS:** No. Not for EGL.

**LV** So they have their own?

**DS:** The last time I looked, they hadn't got that far yet. They can't re-use the EGL hooks directly because the EGL extension is literally: here is a struct for the display, and you can call Wayland methods on it. It's so baked into that that they can't re-use it. In terms of the open source infrastructure, they wouldn't have been able to do that without KMS, much like we couldn't have done Wayland without it, without all the Mesa EGL work, without libhybris, without XKB being common (which I did for Wayland) they wouldn't have keyboard handling. It also took a hell of a lot of work to go into GTK 3 and make it not just an X-only toolkit. Mir re-uses all of these foundations as well.

**LV** Does this mean you'll need separate drivers for Wayland and Mir?

**DS:** There's now an EGL platform extension which now says instead of 'I want to open a display', 'I want to open an X display', or 'I want to open a Mir display', so you can now ship a single blob which supports all of them.

**LV** But it would need to have the Mir bit in it as well as a Wayland bit?

**DS:** Exactly. The Wayland part of that extension is already ratified by Khronos, and that's part of their official set of extensions

**LV** Will 2015 be the year of Wayland on the desktop?

**DS:** I think so. We're already seeing it in set-top boxes because for that kind of stuff where we can say "We can guarantee you super low latency, and no jitter – you can actually use overlays, which you can't in X, that sort of thing". They've already taken a shine to it. I really hope Fedora 21 give the desktop a bit of a kick. It is actually really remarkably solid, at least on Gnome. 