Meeting Notes

> *The Wakefield committers may hold off-line discussions from time-to-time. Since these meetings are just for the committers and invited wayland experts, summary minutes will be recorded here as time permits*

Online Zoom Meeting 9am PDT January 31, 2025

Attendees: Olivier Fourdan, Alexey Ushakov, Maxim Kartashev, Harshitha Onkar, Phil Race, Kevin Rushforth, Victor D'yakov, Alexander Zvegintsev

Alexander:

> *Fixed a problem with focusable popups being closed prematurely, it was not handled correctly before.*
>
> *I'm updating the pipewire header files to a more modern version.*

Alexey:

> *We implemented pixel grabber functionality for Vulcan hardware accelerated surfaces.*
>
> *We're creating a sort of shortcut for the robot functionality to grab pixels directly from it.*
>
> *I implemented it to do single window testing for rendering.*
>
> *Within this work I implemented surface to software blit that is necessary to get the pixels, now running some tests from AWT part for regression tests, investigating the remaining failures.*

Maxim:

> *As you probably all know, there are no absolute coordinates in Wayland. So for getLocationOnScreen I just returned zeros.*
>
> *It worked for a simple Swing application, and it worked perfectly.*
>
> *But it turns out that there are actually cases when you do have some glimpse into the Wayland coordinate system when you look at the relative location of the monitors that you have attached to a system.*
>
> *And in XToolkit, it is visible, it matters, but the applications do depend on this.*
>
> *So if you have two displays that are not perfectly aligned, for example, if the other monitor, not that your main window is on, is above.*
>
> *So your monitor has a positive offset in the absolute coordinate system.*
>
> *And this created a small problem, which I fixed by making all windows returning their primary monitors offsets as their absolute locations.*
>
> *So you might have some non-zero offset, if you have more than one monitor,*
>
> *For example, the monitor on the right would be the normal size, the monitor on the left would be the horizontal offset size, and so on.*
>
> *So having this done, I started going around and finding and fixing all sorts of offset issues, mostly with pop-ups.*
>
> *And it took quite a while because I don't normally use a setup where the monitors are not aligned. But people do, and they report problems.*
>
> *And I started using this setup so that I could catch problems before others did.*
>
> *And the other day, for example, I discovered that the robot implementation actually also depends on this feature, that the specification expects the robot to have coordinates on the screen that didn't matter before, but now they do.*
>
> *One is the test that would just get the location of a JFrame, which is wrong because its specification says that this location is relative to the parent and not to the screen.*
>
> *So to properly feed the robot the correct coordinates, you need to get the location on the screen.*
>
> *But even if you do that in our Wayland toolkit implementation, you may still get the wrong result because of screen offsets in a multi-monitor configuration.*
>
> *So the problem is simple, but it took a long time to figure out which places to fix and how, and I'm sure we're not done yet, as not all holes have been plugged.*

> *As more people start using our pure Wayland toolkit, more bugs are reported, unobvious configurations are discovered, etc.*
>
> *For example, someone recently tested the clipboard and discovered that it would not accept more than 64K of data.*
>
> *And it turned out that at least on my Ubuntu with GNOME 24.04, the pipe opened for clipboard content transfer is in non-blocking mode.*
>
> *And the old IO in Java doesn't handle this very well. So it just exits as if successful.*
>
> *And fortunately after writing the buffer capacity, which is 64K, it continues as if nothing happened.*

Online Zoom Meeting 9am PDT December 5 2024

Attendees: Jonas Ådahl, Olivier Fourdan, Alexey Ushakov, Harshitha Onkar, Phil Race, Kevin Rushforth, Victor D'yakov, Alexander Zvegintsev

Alexander:

> *Fixed a couple of JavaFX/Swing interop issues related to our screencast usage for receiving screen data:*
>
> *1. hang when FX is running, it internally runs the gtk main loop, which wasn't accounted for on the JDK side.*
> *So now it handles 3 cases:*
> ** When there is no GTK main loop running*
> ** If there is a GTK main loop running, but we are not requesting pixels on its thread*
> ** If there is a GTK main loop running and we request pixels on its thread*
>
> *2. crash,*
> *Internally the ScreenCast session keeps open for 2s (both JDK and JFX, and their implementations are almost identical).*
> *When we perform a cleanup to close the session, we internally called pw_deinit before the fix.*
> *It becomes a problem if these sessions overlap in time, so that the second session cleanup crashes when it tries to call pipewire functions without initializing the pipewire system by pw_init*
>
> *Working on JDK side fix of the emulated resolution change does not come issue.*
> *The system side fix has landed on the Ubuntu 22.04 only partially, as it doesn't receive configure notification for change back to the native resolution(other works), but it 24.04 it works with native application as it has both fixes..*
> *It still needs changes on the JDK side.*
>
> *And there are a few more issues to fix.*
> *The ones that were already discussed before like HiDPI support. It was decided that it's hard to implement for x11 compatibility mode. didn't dig into it, so I'll take a look at it later.*
> *There is another issue with drag and drop when we do a drag and drop within a single window when the CTRL key is pressed. I haven't managed to make a native renderer for this yet.*

Alexey:

> *Regarding high DPI support, I would suggest to discuss it with Nikita Gubarkov via email.*
> *Because he spent a lot of time trying to support high DPI in Xwayland and got some basic limitations.*
> *So he even implemented some logic and we tried to make it work in production.*
> *But we found some really hard corner cases and were not able to fix them.*

Alexander:

> *I seem to recall that it had something to do with multiscreen.*
> *So probably we can implement this support at least for a single screen.*

Jonas:

> *For Gnome 47 or Fedora 41, I've implemented support for what we call native scaling for XWayland(for mutter).*
> *It's more or less similar to how one of the KDE HiDPI X11 modes works where we tell XWayland to draw bigger.*
> *And then we tell X11 clients to draw bigger than our logical monitor layout in the compositor.*
> *And then we take each window from the X server and scale it down to the same scale that we told XWayland to scale itself up to.*
> *This works well for all X11 applications.*
> *We have applications that use what we could call traditional X11 high DPI, where the Xserver has a global scale and windows, all different windows in the application try to follow that scale.*
> *But when I was running this with Intellij, it was using X11, but if you had access to a Wayland connection, it dynamically adjusted its scale to whatever monitor you were using.*
> *It was according to the WL output and that caused problems because whatever it was on a low DPI monitor, there was no way for the compositor to know that it wasn't large anymore, and it got very, very tiny.*
> *Is there a way to detect this kind or disable this from a window manager or Wayland compositor direction so that this doesn't happen if you run Intellij?*

Alexey:

> *I suppose that we're going to revert this logic in the future versions of EDA so it shouldn't be a case anymore.*
> *But it's interesting. Do you have some links to read about this feature of XWayland? So maybe we can find out the way how we can use it.*

Jonas:

> *https://gitlab.gnome.org/GNOME/mutter/-/merge_requests/3567#note_2159205*

Alexey:

> *Yes, we have been collecting feedback from our users with the Vulkan rendering preview and we have found some issues with different environments.*
> *Some users are trying to run Vulkan with hardware accelerated WSL2 and we actually looked into it and it didn't work, but we found out why.*
> *That kind of acceleration is not fully supported by the drivers that they a have implemented.*
> *And in particular, it was an unimportant feature, the XOR mode.*
> *So probably if we make the XOR mode optional, we'll be able to run on WSL with hardware acceleration as well.*
> *There were also some issues with the NVIDIA hardware.*
> *We are currently working on all of those issues that we got from our users.*
> *And we're also currently implementing the feature that helps us test Vulkan.*
> *In the Wayland software mode, we have implemented some of the PixelGrabber functionality for a window.*
> *We go directly to the surface of the window and grab the color from there.*
> *So we are able to test at least one window application without the help of the operating system.*
> *And right now we've almost implemented the approach to grab pixels from Vulkan windows.*
> *And I think we'll have an update next week.*
> *And also Maxim fixed some corner cases in Wayland software mode.*
> *Connected with some dialogues and context menus with the multiple monitors.*

Online Zoom Meeting 9am PDT November 8 2024

Attendees: Olivier Fourdan, Alexey Ushakov, Phil Race, Kevin Rushforth, Maxim Kartashev, Jonas Ådahl, Victor D'yakov, Alexander Zvegintsev

Alexander:

> *ImageDecoratedDnD.java now fails on Gnome 47 (Ubuntu 24.10), DnD is aborted when trying to run it with CTRL pressed.*
>
> *It looks like a non-java issue, as it doesn't reproduce at all when mouse and keyboard interactions are done via XTest(java.awt.Robot)*
>
> *The issue is reproducible when an actual user performs mouse and keyboard actions with physical devices (which didn't happen before), sometimes it passes though(maybe 1 out of 20 attempts).*
>
> *Now I am trying to make a native reproducer for it.*

Alexey:

> *Yes, it actually took longer than we expected.*
>
> *In Vulkan we have validation layers and despite the fact that it works visually we had a certain amount of validation errors in the pipeline because of some mis-synchronization and things like that and we actually fixed all the validation issues.*
>
> *They're actually enabled in the fast debug build and you also need to have validation layers installed in the system to make them work.*

> *We are even sending out some preliminary builds to our users and have gotten some good feedback from them recently.*
>
> *The Vulkan support provides the list of the hardware devices(if you enable it with the True with capital)*
>
> *We've got a list of hardware devices that our users are using, some people are using m1 macbooks with Asahi Linux installed, and Vulkan has been working on hardware accelerated drivers for that Linux.*
>
> *So currently we are putting all the changes we have made in our JBR into the separate branch that Maxim recently updated, which is a jdk21.0.1-wayland.*
>
> *Obviously it's not as performant now, but we're working on the primitives.*
>
> *And the ultimate goal is to use an accelerated image for rendering after all the primitives are done.*
>
> *In the future we will work on the renderer.*
>
> *Actually, one of our engineers is working on enabling Vulkan on X11. So it looks like it's a fairly simple change, but it's quite tricky. So maybe in this case we will have Vulkan in both pipelines.*

Maxim:

> *One issue wasn't just a bug fix.*
>
> *It sort of pointed out to me a potential incompatibility between Java and Wayland: Wayland requires proof of user involvement (like serial numbers from user events) for actions like clipboard access.*
>
> *The protocol, however, doesn't specify which events are accepted, nor does it provide feedback, leaving developers to experiment and guess.*
>
> *After extensive testing with various Wayland implementations, I found a solution that works for now.*
>
> *This is extremely fragile and any other Wayland implementation that does things differently will kill the Java assumption and things will start to break.*
>
> *We also got rid of the generated code in our development branch, it is now generated on the fly during the build.*
>
> *You only need to install one extra package, wayland-protocols, which is not hard at all, except for a gGnome-only protocol, which should be brought by hand, as I don't know where to get it on any distribution. I made it optional as it is not essential, it just makes the dialogs work better and more gnome-like.*

Jonas:

> *We recently upstreamed the xdg_dialog protocol, and it has set_modal.*
>
> *And regarding the clipboard and the ambiguity of what kind of serial input events are respected and all that, I suggest opening an issue in the Wayland repository issue tracker to see if it can be clarified. At least to know which parts are expected and which are implementation specific.*

Online Zoom Meeting 9am PDT October 10 2024

Attendees: Olivier Fourdan, Alexey Ushakov, Phil Race, Kevin Rushforth, Maxim Kartashev, Victor D'yakov, Alexander Zvegintsev

Alexey:

> *We have some achievements from the Vulkan implementation.*
>
> *We finally implemented blit support, for blits, which is necessary to actually present content on the screen.*
>
> *So currently we actually have a software rendering mode presented via Vulkan.*
>
> *It's even possible to run our IDEs, for example Idea, in this mode.*
>
> *We have like in a metal texture pool and able to render used textures as a back buffer in Vulkan and present them on the screen.*
>
> *We also implemented some primitives in pure rendering mode, but they do not work now because currently we have volatile image as a buffered image.*
>
> *Now we still have rendering in software mode presented via Vulkan.*

> *We've now implemented high DPI support for this mode, so it's possible to use this hybrid rendering mode with different scale factors.*
>
> *We're going to be polishing these changes and maybe early next week we'll get them into the Wakefield repository.*
>
> *From a performance point of view it doesn't make sense, because this hybrid mode is slower than even the pure software rendering mode in wayland.*
>
> *It's a pretty good step from an implementation point of view, because now we can test our texture pool and drag application windows from screen to screen with a different DPI and see how it reacts and rebuild the texture buffers and stuff like that.*
>
> *We continue to implement primitives to finally use a real hardware accelerated volatile image as a backbuffer.*

Phil:

> *What platforms are you testing on? Is everything equally stable on the Vulkan pipeline?*

Alexey:

> *Currently it only works for Linux and Wayland. We have discussed other platforms, it is quite easy to extend this support to Windows and X11, it is an isolated part of the pipeline.*

Phil:

> *What kind of hardware are you testing on? Raw, VirtualBox or something like that as well?*

Alexey:

> *We tried it mostly on AMD hardware. So we tried desktop cards and mobile cards.*
>
> *Unfortunately NVIDIA is very unstable for Wayland.*
>
> *By the way, Vulkan has its own software rendering pipeline inside, you can actually run Vulkan pipeline in software rendering mode.*
>
> *We've implemented the ability to switch between the different cards and the software rendering pipeline via the VM property.*
>
> *You can get the list of devices that you have in the system, you can run on that particular device in the next run and run on that particular device.*

Online Zoom Meeting 9am PDT September 12 2024

Attendees: Niels de Graef, Olivier Fourdan, Alexey Ushakov, Kevin Rushforth, Victor D'yakov, Phil Race, Alexander Zvegintsev

Alexander:

> *The fix for the DnD issue is verified, everything works fine(including compatibility testing).*
>
> *Relax The java.awt.Robot specification is now shipped with earlier releases such as 17u, 11u, 8u, so the X11 compatibility mode related fixes can now be backported.*
>
> *The fix for taking screenshots is also delivered for Java FX, however it introduced two more issues:*
>
> *JDK-8335468 calling java.awt.Robot#getPixelColor hangs when the JavaFX is running. I have the fix, but it leads to the second issue.*
> *JDK-8335469 Crash when OpenJDK and JavaFX screencast sessions overlap, e.g. getPixelColor calls from FX and JDK robots within 2s (keep sessions open interval to avoid overhead of creating/closing sessions for frequent screenshots). Also have a fix, but it is hard to find an object to sync on in JDK and JavaFX without bringing new extra dependencies between them.*

Alexey:

> *We successfully launched software rendering wayland as a preview in 2024.2 release and we generally get quite good feedback from the users.*
>
> *They switched to our mode and use it on the wayland, but there are still some issues in different parts input methods for some tricky input frameworks doesn't work. But in general it looks like it's the right way to do it.*
>
> *Currently we are fixing some bugs and implementing missing parts in input and continue to work on Vulkan rendering.*
>
> *So we finally got it right and are working on implementing certain rendering primitives and internal stuff. So hopefully we'll get new mode of Vulkan produced in the next release of our products.*

Online Zoom Meeting 9am PDT June 20 2024

Attendees: Olivier Fourdan, Jonas Ådahl, Maxim Kartashev, Kevin Rushforth, Victor D'yakov, Phil Race

Vacation season : several people out today, and because of same we will not meet in July

Phil : DnD bug https://gitlab.gnome.org/GNOME/mutter/-/issues/3511 is a TCK compliance issue for JDK
Jonas : DnD bug should be fixed in Gnome 46.3 (6-29-2024)
We may need to follow up with distros to get it backported.

Phil : is libei ready for experimentation
Olivier: yes

Discussion about c/h files generated from wayland protocol files - we should generate these at build time, not check them in. JDK build will tell a developer of missing dependencies and how to fix during configure. To build on older OSes such as are used by CI build systems will instead require a devkit.
These are already extensively used by the JDK build.

Maxim:
Keeping Wakefield repo up to date with WIP.

Figured out what seems a reliable way to count FPS on wayland.

Have someone to work on Input Method support - common request
Jonas : Is this V3 - also there may be a meeting this summer in an occasional series to discuss requirements for input methods. Maxim will send contact info so the person working on it can participate if/when it happens.

Online Zoom Meeting 9am PST May 23 2024

Attendees: Niels de Graef, Jonas Adahl, Alexey Ushakov, Kevin Rushforth, Victor D'yakov, Phil Race, Alexander Zvegintsev

Alexander:

> *I was mostly focused on stabilizing tests. Many of them were failing, some for the non-obvious reasons.*
>
> *For example, there is a difference. It looks like there is a difference in implementation between XWayland server and regular X11 server.*
>
> *We have a couple of tests where we are not releasing the mouse keystroke or the keyboard keystroke.*
>
> *This looks like the regular X11 server would release these keys when the client disconnects.*
>
> *But for some reason, XWayland does not. For example, we can get a failing test like a few tests after those tests which failed to release this key.*
>
> *But just releasing this key is good enough to fix that.*
>
> *There was some test that counted mouse enter/exit events and making some action to frame like iconify, deiconify, maximize and so on.*
>
> *For some reason after maximizing both for a frame, mouse coordinates reported by XWayland go crazy and we are missing these events.*
>
> *Is it our issue or not? But it seems like a bug somewhere outside.*

Victor:

> *And just to explain to everybody on this call, by stabilizing tests, we are talking about open JDK in the mainline, which right now is JDK 23.*
>
> *So we will fork in two weeks.*

> *So it means at that moment mainline will be 24 and we don't have any plans to work on 23 at that moment unless it's really critical issue.*
>
> *So it means we will continue to work only on mainline and 23 will be for stabilization.*

Phil:

> *So the tests that Alexander fixes that are unstable are the ones that are only unstable when we run them on Wayland.*
>
> *And these are, it's all about just tweaks to the tests, not about tweaks to, not about changes to the product code.*

Alexey:

> *We're currently stabilizing the Pure Wayland branch, Pure Wayland toolkit in our fork, JetBrains runtime repository, fixing some issues to be able to release it in preview mode, not by default for the users.*
>
> *There are some active community members that would like to implement something in Pure Wayland toolkit.*
>
> *Maybe you've seen the recent pull request to the wakefield repository with server-side declaration.*
>
> *We're also working on hardware acceleration.*

Phil:

> *Alexander is going to have to go off and be distracted from the pure Wakefield project focused on AWT in order to help Java FX catch up.*
>
> *Because FX is going to have the same problem that we have had with the JDK.*
>
> *We're not going to be able to run FX on upcoming distros unless we actually fix this.*
>
> *And that will be right after the fork. I already mentioned that in two weeks.*
>
> *But I, I think 95% of what it shouldn't be anything like as hard to do as what Alexander had to do for AWT.*
>
> *I think it should be mostly lift the code, move it over here, drop, integrate, hook things up.*
>
> *So it shouldn't be too bad. I think all the hard learning has been done.*

Alexey:

> *We discussed the high DPI support for XWayland with Nikita.*
>
> *And based on our experience, we decided not to port it to OpenJDK, because it works in some cases, but in general that approach doesn't work well.*
>
> *And we have a lot of regressions and we fix them.*
>
> *And we decided to fix those problems and just move forward.*

Phil:

> *Were the issues with fractional scaling or with integer scaling as well?*

Alexey:

> *If you have monitors with different scales, and if they are placed in a strange position in virtual space, e.g. on top of each other, it may be difficult to calculate the position of the windows, because some information is missing on the X11 side.*
>
> *So it works for monitors that are placed one side of each other.*
>
> *But if you put them on top of each other, it causes some problems with, for example, menus or things like that.*
>
> *Nikita provided the initial solution to the open JDK alias with this fix.*
>
> *We can actually provide the complete solution just to look at it and see if we can reuse it in the XWayland area. We don't have a lot of resources to work on pure Wayland.*

Phil:

> *If you're basically saying the high DPI stuff isn't working.*
>
> *Is it because of issues in, is it just a matter of working through the issues on our side?*
>
> *Or is there something, is there a problem on the compositor side or is there something missing in terms of an API?*

Alexey:

> *Yes, exactly. There's something missing in the API. We cannot get all the information to calculate location in some configuration, some display, multiple display configuration, unless we provide some additional API for XWayland sandbox.*
>
> *It's less tricky in a pure Wayland mode. So we decided to just switch it.*

Alexander:

> *Speaking of RHEL10, there will be removed Xorg session. Will it be possible to add it back somehow?*
>
> *Or will it be removed completely?*

Niels:

> *So there's a there's X11, there's the X11 session and there's Xorg.*
>
> *And so X11 itself is a protocol and will be there in the form of X Wayland.*
>
> *The Xorg session, that will be gone by in the default trial.*
>
> *It could be that that's going to be so I think there will be people who will have an Apple package.*
>
> *So Apple is the repository that people can install, but it's not supported by Red Hat.*
>
> *So people will probably be able to install it, but it won't be supported.*

Phil:

> *OK, so in other words, you can't just go to the Red Hat package repository, pick something, install it, and get Xorg back.*

Niels:

> *No, indeed.*
>
> *I recently looked into some of the Wayland protocols that are still under discussion.*
>
> *So one of the things that piqued my interest was the splash screen protocol.*
>
> *And one of the things I was wondering is if it makes sense to get some feedback from different people and see what they expect from a splash screen,*
>
> *because that's one of the common ones. It's one of the things that we know that people are asking for.*
>
> *For example, for global positioning, they want to have a splash screen that is centered somewhere in the middle of the screen, so it presents the user nicely.*
>
> *So one of the things we could do is have a separate splash window, and a special protocol for splash windows.*

Phil:

> *But the OpenJDK splash screen protocol and its API has couple of different there's some interesting there's an interesting aspect to it.*
> *You can basically say I'm starting Java and you provide a command line option dash splash colon.*
> *And then my pretty picture dot JPEG or GIF or whatever, or PNG.*
> *And that will get put up by slap bang in the middle of the default screen. (Usually)*
> *And it's done by some native code that gets up and running before the VM is even running.*
> *So you can't there's not even anything running Java code at that point.*

In the back in it just forks that off sets runs a thread to do that and then starts them.
That can theoretically be done in a very short amount of time.
And then you still have to boot up this massive thing called the VM and get it to start loading classes and stuff.
Now, once it's got up and loaded classes and all the rest of it, there's a kind of callback mechanism.
And we have splash screen APIs in the AWT.
And there can be a transition for that window to where that window is to actually transition into something where we can then start rendering into it using AWTs.
It doesn't have to be the same window. But the point is, is that for the splash screen to work for us, it's not just it can't just be a fire and forget thing.
We actually need to know where that window is and then be able to seamlessly transition to having an AWT, a normal l top level window.

They're both top level windows, but the first top level window isn't being run by AWT.

It's just a pure native block of that thing on the screen.

No event processing or anything. It's not an AWT window. It's just a native would be a native toolkit or X11 window into which we just call X put image.

Or something. And then but it transitions into this full on Java AWT window in which swing can draw or anything.

And we can put additional information as the application is then starting up.

Another question, what toolkit are you actually going to use, XToolkit or WLToolkit, before you've even run the tool, started to read, run the code that parses the options that decides that.

We really need to know that window, the position of that splash screen, so that we can ask another window to go to the exact same position and size, not just kind of a habit fire and forget. Otherwise, this, the transition doesn't work.

Alexey:

We actually implemented the splash screen in our WL toolkit.

And there were some tricky architectural decisions to decide which window to show before anything existed, the X11 window or the Wayland window.

And yeah, I don't remember the exact logic, but it was a tricky question

Jonas:

The question was, does the splash screen image thingy share the same display connection as the AWT one, but if AWT isn't even started yet, does it need to, can it inherit the connection, or is that even possible if you don't know what kind of connection it's going to be until like after.

I guess from you there is that since it's going to be X11 or Wayland, when the splash screen shows up, it's not possible to know if it's even possible to share the connection because it's going to be X11 or Wayland.

Phil:

You ask a very good question. I would be inclined to guess. There must be a separate X connection, but I'm reasonably sure that with a bit of work that we could actually use the same connection.

It might be that we don't share the same connection, but I'm pretty sure we could.

Jonas:

Is the splash screen always the same size as the window itself? Is that what the purpose is that it shows something?

Phil:

That's the purpose. Yes, that's exactly the purpose.

There's the splash screen that gets loaded initially, where there's, you can't do any rendering into it.

You just basically give it a GIF or a PNG and it's displayed via very lightweight native code.

So the splash screen is fast. The splash screen, the window on the screen that has the PNG in it is like supposed to be instant, like 10th of a second.

And then you got another little bit of time while the VM boots everything up.

> *It could be drawing a progress bar as it basically ticks off loading all the plugins and all sorts of other things that some big application might do, making connections, reconnecting to the server.*
>
> *And then, then generally that window then, most, in most applications, that window is not going to become your application window. It is just , the information we're getting going, we're getting going.*
>
> *And then boom, it disappears because now your main application window is up. And that's the usual way people use it.*

JonasL

> *It makes me wonder what exactly is missing from having a splash screen protocol that gets you the PNG on the screen, more or less instantaneously.*
>
> *Is it that you have to morph it to the next one somehow? Like you need to create an association between the splash and the next window? Or what exactly?*

Phil:

> *I have to go off and double check our API, but to see how the transition is done. The same connection, but I don't think it's the same actual window is the point.*
>
> *It's the same server connection, but it's not the same window. We just replace the window with the new one.*

Alexey:

> *As far as I can remember, in Wayland Toolkit, we actually use the same window.*
>
> *Because it's impossible to create the similar window with the same, the similar position.*

Jonas:

> *With the splash protocol that I suspected would have to be a different window that you would.*
>
> *And if there needs to be some kind of association of one knowing about the next, then that's something to take into account in how you design the protocol.*

Niels:

> *Another question is about making a splash screen output only, which means that basically it would not be allowed to get any input.*
>
> *I don't know what kind of problem that might be for you, or if that would be acceptable to people.*

Phil:

> *The window is definitely "output only", so I think that part fits into the proposed protocol.*

Alexander

> *And one aspect we didn't mention, probably not an issue, is that the splash screen window should support shaped windows.*

Phil:

> *You probably right. because any window can be a shaped window(e.g. a window that looks to you like a circle rather than a rectangle).*
>
> *And generally it's done by applying a clip, essentially, between the implementation and the display server.*
>
> *And there's generally some kind of transparency involved to help make it work. And for the initial image that you're displaying as a PNG, if you're displaying a JPEG, which doesn't support transparency, you can't.*
>
> *But PNG and GIF could actually have transparent pixels and some clever, some very clever developer could basically craft their nice PNG, which is 500 by 500 or something.*
>
> *They know exactly what part of the window is transparent and what part isn't.*
>
> *And then they have to apply the same, apply a clip to create the AWT window when it comes up that matches that exactly.*

> *Otherwise you won't have a seamless transition.*

Alexander:

> *It should also be input transparent, not just visually transparent.*

Jonas:

> *Something to keep in mind about the input region, maybe not relevant for splash screens, but if you have a shaped window that only wants to receive input on the shape, the representation is trivial. It's just a texture or a buffer with an alpha channel, which is supported without any problems.*
>
> *The problem is that the input region is a region. This means a set of rectangles.*
>
> *So if you have smooth lines, like bezier curves, that's going to be a lot of regions, a lot, a lot of rectangles. So it's very inefficient.*

Phil:

> *The only saving grace is shaped windows are, not many people use complex shapes for, in practice for, it's it's the usual sort of thing where some nice curved, some nice curved corners or something, right.*
>
> *But it's, which means that maybe the problem is not quite as bad. You don't see many real user interface applications that have a some kind of a complicated thing that looks like a Mobius strip as your, as your window or something now that's not exactly. I would say it's almost nonexistent.*

Online Zoom Meeting 9am PST Apr 25 2024

Attendees: Niels de Graef, Kevin Rushforth, Victor D'yakov, Alexey Ushakov, Olivier Fourdan, Phil Race, Maxim Kartashev, Alexander Zvegintsev

Phil:

> *We made changes in the specification of the Java 21 to accomodate Wayland.*
>
> *At that time wasn't expected to be backported, but in order to be able to run 8,11,17 which are LTS and will be supported for a significant period of time,*
>
> *that will overlaps with the lifetime of RedHat Linux 10 without the X11 session support.*
>
> *Maintenance Releases of Java 8, 11 and 17 specifications, will have these specification changes to allow the implementation update on those.*

Alexey:

> *Do you plan to port all the code that we are going to write to that releases? Pure Wayland Toolkit?*

Phil:

> *Not a chance. This is purely by XWayland, this is a spec change, we not even backporting the Robot changes we did in 21.*
>
> *At some point RedHat, Oracle, Azul or somebody else might decide to put the Robot changes into their versions of 8, 11 or 17 as well.*
>
> *After the specification changes in MR, you don't have to backport Robot changes, you are compliant at this point, but it just doesn't work very well.*

Alexander:

> *I've been busy with beforementioned spec backports for the MR, discovered and fixed missing doPriviliged calls on our TokenStorage implementation,*
>
> *Fix tests that try to interact outside the XWayland server. I also tried to test the XTEST - libei support(should be available on Gnome 45) on beta of Ubuntu 24.04(Gnome 46), but it doesn't work, no confirmation dialog appeared.*
>
> *I tried the xdotool and java.awt.Robot(XTEST internally). Is this supported only on Fedora?*

Olivier:

> *just a couple of comments on xdotool, it heavily using XTEST for everything, I personally fixed upstream, it might now work.*
>
> *I believe mutter had support of libei before XWayland, this is still optional, meaning that it depends whether XWayland itself have been build with libei support or not.*
>
> *So it depends how XWayland was build on Ubuntu, but if they build the XWayland with libeie, than the XTEST would go through the libei as you expected.*

Alexander:

> *Another question, what about the HiDPI support patch for the XWayland? It's probably good time to make it into JDK 23*

Alexey:

> *We actually switched to the pure Wayland implementation now, but I ask Nikita to continue work on this.*
>
> *I continue my work on Vulkan implementation, hopefully we will have something next week. It is similar to the Metal work, but this API is more complicated.*
>
> *I have something working, but I need to put it together to at least have something interesting to show.*

Niels:

> *Last meeting you mentioned that software rendering practically complete. It is interesting to know how people evaluate this.*

Alexey:

> *Software rendering works quite good, but we got everighing possible from it. It is slower than hardware rendering. From our perspective is quite good for the preview.*
>
> *In some cases we have low performance.*
>
> *Maxim can provide more details.*

Maxim:

> *We have received quite a few reports from users who tried this Wayland Toolkit and run our IDEs on that, discover some bugs.*
>
> *Actually nobody complains about performance, I don't think that expectations are high, people know that it is not hardware accelerated.*
>
> *Main complaint was from, I accidently discovered that window resizing was way slower on Wayland Toolkit comparing to XToolkit.*
>
> *I made a few changes and now we got ~30 fps in 4k when it was ~10 fps before, it's quite smooth, even with software rendering.*
>
> *We made a bunch of fixes here and ther since last meeting, e.g. one of the guys added support for more than 3 mouse buttons, which was not present before.*

Olivier:

> *There is a proposal for a new category of Wayland protocols at compatibility. That could be useful for Wakefield project as well. Right now it is more about creating the category.*
>
> *The reason why I wanted to mention it, somebody mentioned Wakefield as part of the discussion, specifically about something Wakefield does, about a bug with the maximum size of a window.*
>
> *Is that something that we know about?*

Maxim:

> *If understand correctly, I did a small change relatively recently with regards to limiting the maximum window size, because one of the tests started crashing JVM. It creates a window of size MAX INT.*
>
> *Now it limits the size twice the size of combined monitors.*

Online Zoom Meeting 9am PST Feb 29 2024

Attendees: Jonas Adahl, Kevin Rushforth, Victor D'yakov, Alexey Ushakov, Olivier Fourdan, Phil Race, Maxim Kartashev, Alexander Zvegintsev

Alexander:

> *Not much to report, continuing my aarch64 testing on Wayland, so far so good.*

Maxim:

> *Very minor update from my side, just mostly bugfixes and small changes, people continue to use our unfinished toolkit on various Linux distros and report lots of bugs and wishes.*
>
> *For example, many seem to be bothered by the mismatched title bar thing and stuff like that.*

Alexey:

> *We have solved all performance issues with our runtime based on JDK21, so we will release JBR21 runtime based on it inside our product with 24.2 release.*
>
> *It'll be bundled with Wayland support, ofc is not by default, we may have more testing within our users.*
>
> *There is some unofficial testing of pure Wayland toolkit within our community. I am personally working on CLion and Idea running on top of Wayland, of course we have some issues, but it generally works.*
>
> *Even in software mode it has quite good performance.*
>
> *I also continue to implement hardware accelerated rendering using Vulkan, just got some basic rendering working.*

Phil:

> *Have you guys tested the arbitrary shaped windows support?*

Maxim:

> *Yes, the ruler example works, it is actually semi-transparent by the way.*

Online Zoom Meeting 9am PST Feb 29 2024

Attendees: Niels de Graef, Kevin Rushforth, Victor D'yakov, Alexey Ushakov, Olivier Fourdan, Phil Race, Maxim Kartashev

Phil & Alexey : FOSDEM Free Java room session on Wakefield went well.
Session can be viewed here : https://fosdem.org/2024/schedule/event/fosdem-2024-2154-openjdk-project-wakefield-the-wayland-desktop-for-jdk-on-linux/

Alexey : Not much to report, been occupied with other tasks

Maxim : Ramped up testing so ran all AWT + Swing groups.
With WLToolkit - 70 % of Swing tests passed and about 25% of AWT tests passed
Trying to run in a weston instance via plugin which supports setting location, pixel grabbing,
input events. First obstacle was the stability of weston window manager which crashed in 10 mins.
Have just implemented a special mode in jtreg which would run each test in a new weston instance

Niels: Perhaps this helper daemon (written by Olivier ?) could help
https://gitlab.gnome.org/ofourdan/gnome-ponytail-daemon

Phil : to Neils: Is there any update adding restore token support in RHEL 9.x
Neils : yes, https://issues.redhat.com/browse/RHEL-4526 - shows it is targeted to be fixed in 9.4

Maxim :
Their users when learning that wayland doesn't support setLocation for window positioning are not happy.

Phil: whilst we have screencast for pixel grabbing, libei upcoming for events,
positioning is not something we know the answer to yet. Yes, spec. allows for it,
but it is a practical problem.

Niels: there are proposals around protocols for specific cases
Eg splashscreen protocol but not much upstream interest so far in implementing it
The session management protocol is also proposed
https://gitlab.freedesktop.org/wayland/wayland-protocols/-/merge_requests/18
and there is at least some work going on there :
https://gitlab.freedesktop.org/wayland/weston/-/merge_requests/1444
Picture in picture has its own protocol used by firefox

Suggest go to upstream discussion groups and try to have productive discussions with
But a core API for setting global positioning is a hard pass for the compositor,

Maxim : session mgmt might help Intellij

Maxim: What are Oracle's plans for Wayland
Phil : Finishing up Xwayland support in JDK 22, there's also a need to add support
for XWayland to JavaFX's robot class. Sometime after that we will be able to turn attention
to pure wayland but OpenJDK projects don't have specific deadlines. When they are ready
we can have a JEP to propose to integrate it, but it has to be 100% TCK compliant for that.

Online Zoom Meeting 9am PST Jan 4th 2024

Attendees: Niels de Graef, Kevin Rushforth, Victor D'yakov, Alexey Ushakov, Alexander Zvegintsev

Alexander:

> *In December I did some testing regarding the "automated tests become manual on OL 9 because of confirmation for each screencapture".*
>
> *How many user interactions (select display, click share screen button) are required to do this?*
>
> *The answer is ~11k, so we are definitely not going to do this and will wait for the restore_token backport to 9.4.*
>
> *Regarding not getting a displayChanged event after a setDisplayMode call:*
>
> *On Ubuntu 22.04, we get a ConfigureNotify for the root window when we switch to a non-native resolution, but we don't get it when we switch back to the native resolution.*
>
> *On Ubuntu 23.04 and 23.10 we get both notifications.*

Alexey:

> *We are continuing our internal testing of Intellij Idea on JBR 21 + pure wayland prototype, so far so good. It is only software rendering for now, but for the IDE purposes it works pretty well.*
>
> *Also, after our discussion with Phil,, we are rewriting our hardware accelerated code for Vulkan in C, it is less fun than it was before, but we are moving forward with this approach.*

Niels:

> *Alexey, when you mentioned the software rendering, did you get things working?*

Alexey:

> *Yes, it works fine. We use Wayland surfaces, and our java2d C loops to render geometry and stuff like that, Marlin for anti-aliasing rendering. It is able to support all the features that we have in Idea.*
>
> *The funny thing is that it works better than X11 code.*
>
> *We want to launch internal and external preview with non-default Wayland toolkit, provide proper VM options for some people who want to test it.*
>
> *We already have some people from our tracker who have built our branches and launched Idea.*
>
> *Of course there are some missing features, for example, we don't have some special shaders for better text rendering, but grayscale antialiasing works quite well.*
>
> *I suppose that most of the work was done by the Wayland server, we present the stuff and in the actual Idea we don't need to pass rendering with hardware acceleration.*
>
> *Probably with some crazy animations it would suffer.*

Niels

> *I didn't expect such positive surprises at the beginning of the year. If it is necessary, I can ask people from Fedora community to test it. But I do not know how far we are from that.*

Alexey:

> *It is not in the main OpenJDK repository, we do some regular syncs with the wakefield repository. And yes, it would be interesting, but it will probably require some manual patching of the distribution you provide.*

Niels:

> *That's where RPM Fusion could come in, so if people want to test it, here's a random package you can install to see if it works for you, and if it doesn't you can go there and file a bug.*

Alexey:

> *Sounds good, we can provide a current branch or prepare a special one in the wakefield for this purpose.*

Online Zoom Meeting 9am PST Dec 7th 2023

Attendees: Niels de Graef, Victor D'yakov, Alexey Ushakov, Philip Race, Olivier Fourdan, Maxim Kartashev, Alexander Zvegintsev

Alexander:

> *I don't have much to share regarding Wakefield, just that the fix for not showing focusable popups has been pushed.*

Alexey:

> *Recently Nikita asked about usage of C++ standary library in on Wakefield mainling list, I understand is not quite good for us. What about just C++? Hotspot use it.*

Philip:

> *I do not like C++ at all, hotspots themselves had consequnces it terms of they basically had to abandom supporting some OSes because of C++,*

> *and the Solaris organization said that they deliberately never allowed any part of core solaris to be written in C++. It just takes away a problem if we just stick to C.*

Alexey:

> *The question is whether we should use C heavily or just move to java where it is possible?*

Philip:

> *Yes, that sort of the architechture we've had in every other implementation of everything, it all lives in java then you just call native.*
>
> *If we were to look at a long term future, we could use FFM. The idea there is that you write everything in java, and the native library just accessed throught FFM.*
>
> *So writing it oriented around the native implementation, makes it a little bit harder to move to the FFM.*
>
> *So I am not saying that you should adopt FFM for this, but you might want to think of it.*
>
> *FFM is final as of JDK 22, I already pushed a small usage of it, but I have to jump through hoops. The biggest issue that we have that it needs some time to warm up.*

Maxim:

> *The largest chunk of work that I did was to try to merry code in the Idea to the new reality of Wayland, some quirks that I asked on the mailing list,*
>
> *when some window loses focus and it cannot know which one is going to get it.*
>
> *The approach of waiting just a little bit for the other window to aquire focus works quite well(so far in a very limited context), but I am still trying things out.*
>
> *Philip:*
>
> *We saw that the Red Hat Linux 10 is planning to drop the X.org and X11 session, do you know when Fedora is going to do so?*

Niels:

> *Since Fedora is a community project, it is up to the community to decide. Fedora can take many more packages, if they want they can have a Cinnamon or other sessions, but we will not have X.org in RHEL10.*

Philip:

> *I have a question regarding the OL/RHEL 9, Red Hat empasized stability, but the version of Gnome doesn't seem to support the restore token functionality that we make important use of it the screen capture code.*
>
> *It was added in later version of Gnome, which did not make into 9, and it seems to be dooming us never be able to test Red Hat Linux 9 in X11 compatibility mode. If is definitely the case that they won't be ever upgrade Gnome in 9?*

Niels:

> *We usually keep packages the same, but we do backports when required. Specifically for the restore token I thought we solve this in 9.3(released few weeks ago). But if it isn't there you can always file a bug for that, and we can internally investigate it.*
>
> *My advice would be to check the latest available.*

Online Zoom Meeting 9am PST Nov 9th 2023

Attendees: Niels de Graef, Victor D'yakov, Alexey Ushakov, Kevin Rushforth, Philip Race, Olivier Fourdan, Maxim Kartashev, Alexander Zvegintsev

Alexander:

> *We have a regression for my fix for a popup dismiss on focus lost. It was reported by an external developer.*
>
> *My old fix does not take into account that popup may be focusable(e.g. has TextField). I have a solution and will publish it soon. And we'need to backport it to JDK 21.*

Alexey:

> *We are making a good progress so I will let Maxim give more details about on that, I am more focused on rendering, there is not much progress there.*

Maxim:

> *Most of the progress was not in the wakefield framework, we started experimenting with running our IDEs using the pure Wayland toolkit, so far so good. I had to make quite a few modifications to the IDE code that relied on certain XToolkit related classes and made a lot of assumptions out of that. There are, of course, there are quite a few hiccups here and there.*
>
> *For example, some of the popups just flash instead of showing themselves at a certain position on the screen, or pasting does not always work for some reason, it starts choking every 5 times. There are weird corner cases and I'll have to go through mostly the IDE code, rather than the toolkit code to iron them out before we go public with this.*
>
> *It is still usable, performance wise feels about the same as XToolkit. Resource wise, it seems to consume slightly more memory, but I haven't spent too much time looking into that.*
>
> *My main concern for me at this point is a window placement, and I was wondering if guys from RedHat might have some advice on that. I discovered that gnome has a kind of extension interface for Wayland, where you can set a modality for your window. It has a very nice effect on modal dialogs, it darkens the background, it does not allow to it does not allow the focus to be transferred to the parent window, it shows the dialog centered on top of the window that it is its parent, it is all very well, except that many of our windows are not really dialogs, although they act like ones. For example, the find window, it is not modal, it does not prevent you from focusing a window underneath it, and the problem for me, and I suspect for many users, is going to be that it always opens towards to the top left corner of the screen no matter where you opened it last time.*
>
> *So I was wondering if there is any way to at least open the window in the same position it was opened before?*
>
> *Is there anything else I can do, like give the window manager a hint that I want it centered relative to my parent window?*

Philip:

> *I played with the stuff you pushed into the wakefield project repo, it was clearly software rendering. And it is pretty good, reasonably fast, so it doesn't feel like it slow at all.*
>
> *But there are some other things that are noticable, there was not gtk integration, or gnome integration, so the font were all black and white, because it wasn't picking up any of the desktop settings. The positioning thing was very obvious. The exceptions and messages continually being streamed to the console, but the overall UI was fine.*
>
> *Regards to the windows positioning, clearly we have a modal and non-modal dialog, and it would be nice if non modal dialog could get the same positioning and treatment as modal dialogs.*

Niels:

> *As far as I know, Wayland does not provide any kind of positioning, as you already know. Definitely for modal windows, transient for kind of relationship.*
>
> *It is not GTK that decides to put one window on top of the other, it is actually the Gnome compositor, and I think, for example, a tiling compositor may decide not to have modal windows, since it may not have a concept of it.*
>
> *At this point there is no way to specify that one window should be on top of the another.*

Maxim:

> *I understand that it is in general it is an unsolvable problem, as you said for a different kinds of compositors, for some of them the whole concept just does not make sense, but I think I saw some talk about maybe letting the window position to where it was last dissapered to at least give the user that consistency.*

Niels:

> *The xdg-session-management protocol is under discussion and it has not yet been merged. Basically, it allows a client to request that client windows be restored to the same positions as before.*

Philip:

> *We talked about the session manager in the past quite a while ago. I think we need to understand how we actually distinguish Java applications. It doesn't help with the first time, and even if this is absolute positioning, this window that I can recognize comes up with this position, but you really want it relative to your main window, and you window can may be moved around. It is not the something that happens only at startup, e.g. how to recognize a window shown before is the same window shown after, maybe the session manager takes care of all this in some clever way, but I am not sure about it.*

Maxim:

> *First of all the problem of restoring a session and the problem of restoring a window position within the same session are slightly different.*
>
> *For the latter, if you just want a window to appear where it was last time, and you haven't closed your session, I would be satisfied if a window manager would restore a window that has the same parent-child relationship with another window, and the same size as the last time this happened, but I'm dreaming.*

Niels:

> *It makes sense for some compositors, not sure if we do that. I think the Mutter already does some restoration things already, like when you unplug it and plug your external monitor back in, it also tries to do smart things.*
>
> *At this point, the modal dialog is probably the best choice. The Gnome side has something centered in the middle, but it may not work across compositors.*

Olivier:

> *I wanted to mention that the feature that centers modal dialogs in the center of its parent window is actually a notion, that can be disabled by user, so we can't rely on it, because some people just like me have disabled it. And it is not a standard, other compositors (e.g. KDE) may not be doing that.*

Maxim:

> *Speaking of KDE, it worked by default, it tends to center a child window on top of the parent anyway. And it even centered our splash screen, which is essentially a borderless window. I understand that we should not rely on this all the time, but it is as good as it gets, if it works most of the time, or almost always is good enough for me.*
>
> *Speaking of popups being different, there was a curious problem with popups that I ran into, while making the IDE work under Wayland. One of the most popular dialogs in the IDE is the search everywhere which comes up with double shift tap. It is actually implemented as a popup, and while I was implemented the ability to move it on the screen, with popups you cannot ask the compositor to do that for you, but you do have the ability to do it programatically. There is a curious problem with the interactive movement, when you move it together with your mouse, the window moves and the relative position(to the window that just moved) of the mouse also changes, it does not follow your mouse, it just jumps very erratically. You have to be very slow and careful to grab it.*
>
> *Looks like there is no solution for that, other than to remake it with a proper dialog.*

Alexander:

> *At the last meeting we raised a question about getting the Nikita's HiDPI fix into the mainline, is there an update on that? We have less than a month to get it into JDK 22.*

Maxim:

> *If I understand correctly, just yesterday a problem was discovered with this fix, or rather an opportunity to improve it.*
>
> *An XFCE user discovered a problem with window positioning, and it is quite persistent.*

Online Zoom Meeting 9am PDT 12th Oct 2023

Attendees: Niels de Graef, Victor D'yakov, Phil Race, Olivier Fourdan, Maxim Kartashev, Alexander Zvegintsev

Alexander:

> *The fix for taking screenshots on a displays with different scales is integrated.*
>
> *It looks like we will support X11 compatibility mode in the next JDK release(at least for Ubuntu 22.04 for now). Probably we'll also include the latest Oracle Linux and RHEL.*

Phil:

> *Let's try to test the latest Fedora as well, I know it is not on Oracle supported Linux list.*

Alexander:

> *I recall that a fix for HiDPI support in X11 compatibility mode was posted to the Wakefield mailing list for the Wakefield sandbox, I think it is a good time to integrate it to mainline(JDK 22).*

Maxim:

> *I think it was dependent on some other minor thing that I did, but I also don't remember the fate of it. I'll check on that, if the dependency is in the mainline, I'll ping the guy responsible for the fix.*
>
> *Actual rendering with Vulkan started to happen, there is some progress in this area, but not very visual.*
>
> *I did some minor fixes here and there in the Wayland prototype, I implemented the clipboard support, I started working on DnD, but it turned out that is a lot larger that clipboard support.*
>
> *We are trying to wrap things up to come up with a more or less working solution to show the JetBrains IDE, actually at this point, the goal is almost on the side IDE,*
>
> *they do some funny stuff with popups, some of them show up in weird places. It is practically the main major thing that prevents IDE from being usable from pure Wayland toolkit.*
>
> *Tried to run this thing on KDE, but faced some KDE issues, like XServer is crashing every several minutes, unable to logout without killing something, etc.*
>
> *We also have some difficulties to setup out testing virtual systems in the cloud to start with Wayland session, no matter of the config, investigation is ongoing.*
>
> *The amount of tests is not significant, I managed to choose 400 hand picked tests, that still run in pure Wayland session, we have to use som jtreg 7.3 to pass all required env variable.*
>
> *Those 400 are picked that can be used unmodified, so don't use Robot, don't rely on positioning of a window exactly, etc, but they still make sense.*
>
> *I have a question related to splashscreen, how do we implement the decision which toolkit to use, especially when the splashscreen to be shown?*

Alexander:

> *We still have the positioning issue for Wayland windows(we can't contol their position), we can use the X11 splashscreen since it doesn't have one(for now). Even if the toolkit will be Wayland later.*

Maxim:

> *The positioning problem is sort of solvable with a large transparent window to the size of the screen, and then draw you splashscreen inside of it, but the problem of having an X11 splashscreen*
>
> *and then proceeding later with the Wayland toolkit that you have to have the ability to draw over the splashscreen once toolkit get loaded.*
>
> *You have to have a handle of a window that you showed during the startup, and then that handle has to be reused somehow when you have your AWT,*
>
> *when you can paint and everything else, so it has to stay within one toolkit.*

Niels:

> *Can we just tear down a splashscreen window and then immediately start a new one?*

Maxim:

> *It is not technically possible, the shared code, the infrastructure that works on all the other platforms is set up such that at a startup you fire up a window*
>
> *with the ability to save its opaque handle to the window for the code that comes later and be able to draw on that window, other than a static image that we took from a file(jpg, png gif).*

Phil:

> *If we tear it down, at least there'll be some kind of flicker.*
>
> *For now, if you want just to test this out, if this in production, I would stick for X11 for a while, and I only switch to the Wayland implementation where there is a high level of confidence.*
>
> *But you obviously want to test out the Wayland implementation, so I would just use an environment variable.*

Maxim:

> *That is the only thing that touches shared code. Do you have some recomendations what's not to do in shared code?*
>
> *Because this platform is so specific, so it has two toolkits.*

Phil:

> *For now just do what you have to do in the shared code, and we refine it later.*

Niels:

> *Just wondering in regards to CI problem that you have, so one thing you could do is just run mutter or even gnome shell as headless.*
>
> *That you can use to immediatelly run it as Wayland compositor.*
>
> *gnome-shell --headless --wayland --virtual-monitor 1920x1080*

Online Zoom Meeting 9am PDT 17th Aug 2023

Attendees: Niels de Graef, Victor D'yakov, Phil Race, Olivier Fourdan, Maxim Kartashev, Kevin Rushforth, Jonas Adahl, Alexander Zvegintsev

Alexander:

> *As suggested, I implemented the approach when the session is not closed right away, but after a few seconds of inactivity. It gave a significant boost for certain scenarios, e.g. taking pixel by pixel in 50x50 area took 375s before the fix and only 28s after the fix.*
>
> *Other than that working on various HiDPI scenarios for taking screen data in X11 compatibility mode.*

Maxim:

> *We have made some progress since we last met, Vulkan rendering pipeling work is ongoing, we are actually able to render a colored rectangle on the screen, still the bulk of the work was setting up the environment to be able to start making progress, it is almost done.*

> *We also implemented input support for testing through our Weston plugin, so the robot can generate keyboard events, mouse clicks, including absolute positioning, etc. We are getting ready to start running tests in our environment, but we are not there yet because the rest of the toolkit is not ready to start testing.*
>
> *I finished working on tooltips, menus, now all the stuff works more or less, because the positioning left for the Wayland server entirely with only hint that we can give. It works pretty well for us, but things may change in the production, with all variety of desktop configurations.*
>
> *I am finishing the clipboard support and plan to get to work on DnD soon.*
>
> *I think, that primary focus for our work in medium term is going to be Vulkan and getting the IDE window up on the screen and interactable to show our users the progress being made.*

Phil:

> *The hints you refer to positioning are a bit vague, did it go where you wanted it to go, you don't know it went there, but it actually did go there?*

Maxim:

> *It is probably the most complicated part of the protocol I have dealt with so far. I can post a link to the protocol interface, called the Positioner, but it will not make things clearer.*
>
> *In short, you throw away all the calculations you did in the Java code, except for the coordinates relative to the top left corner of the parent window, and then you should suggest to Wayland what to do if that particular position is not suitable, for example, if it is between screens, on the edge of the screen, off the screen. I think this logic in Wayland is quite similar to the Java code we have.*

https://wayland.app/protocols/xdg-shell#xdg_wm_base:request:create_positioner

https://wayland.app/protocols/xdg-shell#xdg_positioner

> *The Wayland API does not allow you to use the results of calculations made by common Java code, because the result of this code is in an absolute position on the screen.*

Phil:

> *So this is a mismatch between what Wayland expects and the way our code has been written historically?*
>
> *There's nothing fundamentally wrong with either side, it's just that they come from two different worlds and it's going to be hard to make them meet.*

Maxim:

> *Yes, basically the work that we do in Java right now for positioning and popups, all that work is done on the Wayland server side, and you can only nudge it in the right direction that you think is preferable for you by giving hints.*

Phil:

> *There is something about you posted on a Wakefield mailing list about libei. Could you please elaborate what are doing there?*

Maxim:

> *The idea for our testing so far has been to rely on a plugin for Weston to use the Weston instance and run tests there. The only ramining part that was not implemented was the input emulation.*
>
> *One of the approaches to achieve that is to use more or less standard way to use the lebei. I asked if Weston supported it so we could just implement it through libei. But it will not be supported, so we decided to use another approach.*

Niels:

> *people might be interested in Peter Hutterer's GUADEC talk about libei: https://www.youtube.com/live/dVVyokoQl2k?t=15312*
>
> *He is the maintainer of libinput and the author of libei. It is a good talk about input capturing and what libei is.*
>
> *Gnome 45 will be released with libei support in a less than a month.*

Phil:

> *X11 compatibility mode mostly seems to be pretty stable, the only issue we have is related to the Meta. disable_unredirect_for_display(global.display) workaround. It seems to work.*
>
> *I did some pretty exaustive testing, we can run thousands of test with screencaptures, whatever else, the token does seems to remain valid, unless we run this test that does the fullscreen capture and everything goes south.*
>
> *Is there a way to do this Meta.disable_unredirect_for_display(global.display) in some kind of startup file?*

Jonas:

> *You can try to use an extension(basically a js) which loads with a system*
> *https://extensions.gnome.org/extension/1873/disable-unredirect-fullscreen-windows/*
>
> *In Gnome 45 and xdg-desktop-portal 1.18 you also get the same token behavior(reuse token and restore session) as in Screencast for Remote Desktop, so you can do a remote control.*

https://github.com/flatpak/libportal/pull/114

Olivier:

> *For X11 compatibility mode, XWayland has direct support for libei, so XTest calls will call libei internally. This allows to control mouse and keyboard outside the X11 server, which is not possible today.*
>
> *For example, clicking on the title of a window to move it to the front should work.*
>
> *However, it does not use tokens, but a dialog for user confirmation.*

Alexander:

> *Is this permission a one-time thing that lasts forever?*

Olivier:

> *No, the XWayland will keep the connection for some time, if there is no activity from the client for a certain amount of time the connection will be closed and a user will be asked again.*
>
> *It does not use the token. From the Wayland point of view you never know if some legit X11 application or a rogue client is trying to hijack your pointer.*

Alexander:

> *What if I need to keep this permission between reboots?*

Olivier:

> *In this case, you would need to use Remote Desktop Portal with a token and use libei directly, not XWayland.*

Phil:

> *For native Wayland, if we use libei directly - we are good, but for XWayland it gets better only when user is there and can click on a session.*
>
> *It is not even a login session , it is XTest - libei behind the scenes session.*

Maxim:

> *I forgot to mention one more thing, the splash screen support is also finalized.*
>
> *I managed to get it to the screen center, the idea is to show a screen sized transparent window and position against it and hope that the Wayland will place the window exactly matching the screen size.*
>
> *I am not sure how well it will work in a multi-monitor configuration.*

Jonas:

> *There may be some glitches, such as animations showing windows appearing, it will not work well with tiling setups.*
>
> *It does not look like a future-proof solution.*
>
> *The proper solution for the splash screen is to work on the splash screen role window.*
>
> *https://gitlab.freedesktop.org/wayland/wayland-protocols/-/merge_requests/140*

Alexander:

> *We could use a X11's splash screen implementation for a while, it does not depend on XToolkit, and does not have such issues with positioning.*

Jonas:

> *It will be problematic in many compositors that use Wayland to do coordinate systems in a compositor. In Gnome terms, this is called fractional scaling mode, and it may cause the splash screen to be blurry.*

Online Zoom Meeting 9am PDT 22nd June 2023

Attendees: Victor D'yakov, Alexey Ushakov, Phil Race, Kevin Rushforth, Maxim Kartashev, Olivier Fourdan, Alexander Zvegintsev

Update from Alexander :

Screencast support for AWT Robot was integrated into JDK 21

Mostly in good shape.

Some issues, failing tests with

- non-default UI scale

- resolution other than native

- tests which gain focus / raise window by mouse click

- some crashes - not related to screen cast (that one is fixed)

- still need to fix some issues

Phil -

Crash seen now is in GTK 3 when using file dialog - not seen on X.org, but probably our bug

Still losing token sometimes, although worked through lots of confusion and found that CI

system inside Oracle periodically cleans out ~/.java where the token was stored

Temp. change to store it in ~/.awt until that is resolved.

Still see token lost sometimes - unclear why.

1) A tight loop invoking a small JDK app that uses Screencast-based Robot code

was able to run 64K times (ie 64K JVM invocations) - before some unrelated XIM leak caused a hang.

So not obviously the number of invocations

2) Number of reboots ? We reboot CI test systems after each task

3) Something done by one of the tests that invalidates it ?? - perhaps for the rest of the session ?

Dividing up full battery of tests and running the subsets over and over seems so far to have not

provoked invalidating the token. It was expected that this would show the problem and narrow down

the cause but none the wiser yet.

Olivier

May have a workaround for the full screen catpure failure mentioned above

Alt-F2, then type "lg", to invoke looking glass.

then in the evaluator type

Meta.disable_unredirect_for_display(global.display)

It immediately takes effect for the current session.

If that solves it might be an ubuntu specific issue

[UPDATE: this did indeed cure it : tracked as JDK bug https://bugs.openjdk.org/browse/JDK-8310333]

More on this at :

https://extensions.gnome.org/extension/1873/disable-unredirect-fullscreen-windows/

possibly the gnome extension might disable it for all sessions

Alexey :

Tried the JDK 21 robot screencast code with their hidpi fixes and found some issues with multi-screen setup and also found some issues with hidpi solution.

These seem to be problems in their code not in the JDK code. Working on it.

Also working on supporting hw accel in the pure wayland branch

Online Zoom Meeting 9am PDT 25th May 2023

Attendees: Victor D'yakov, Alexey Ushakov, Phil Race, Kevin Rushforth, Maxim Kartashev, Olivier Fourdan, Jonas Adahl, Alexander Zvegintsev

Alexander:

> *we have pushed several changes related to the X11 compatibility :*
>
> *JDK-8307779 - Relax the java.awt.Robot specification*
>
> *JDK-8280993 - [XWayland] Popup is not closed on click outside of area controlled by XWayland*
>
> *CSR JDK-8307456 approved for the java.awt.Robot taking screenshots issue wth Screencast. This fix is also targeted to JDK 21.*
>
> *Drag and Drop does not work in java -> wayland app direction: while this DND fix has been released on the system side(1, 2), changes are still required on the JDK side.*
>
> *Our code it quite picky for the drop target, and requires it to be a toplevel with WM_STATE property, whereas the Wayland server uses dummy windows without this property set.*
>
> *I already have a fix for this on the JDK side that I want to put in 21, but first I want to make sure I didn't break anything. In any case, it will be active when the Wayland session is active.*

Phil:

> *We still have test failures, some of them are tests that really was not so stable, and the timing thing on Wayland vs Xorg makes them fail.*
>
> *Interesting categories of failures are number of crashed that seem to be in pipewire code.*
>
> *It is a mixture on the thread that attached to the JDK where we actually calling something and occasionally some threads that pipewire kick of on its own.*
>
> *We have seen at least one that says "stack smash", terminating.*
>
> *And there is one that is not a crash, with one particular test on particular systems, which we still need to analyze, related to be not able to get the screen data.*

> *Some of the crashes are in tests that do a lot of captures in a loop.*
>
> *There is one test which repeatedly moves a mouse cursor and taking a pixel color at mouse position. It makes around ~11000 of getPixelColor requests, which takes no time on Xorg, but can time out with the new API. Alexander said that one request can be 3000x slower on his machine comparing to X11 implementation. So anything that captures pixels in a loop over and over again is a potential problem.*
>
> *Sometimes when a test fails, test harness makes its own screencapture from another VM, so we wondering about the concurrency of all these libraries we are using now.*
>
> *It could be a mixture of bugs in our code, bugs in pipewire that are fixed in a later version, or bugs are not yet fixed.*

Jonas:

> *Some comments about threads, I would first check that you have handled the processes callback from pipewire, and also check that you have not enabled the real time pipewire thread(It is normally not enabled, as you should explicitly enable it).*
>
> *Regarding performance, you are using the screencasting not for screencasting, not the way it is intended, so it is not strange that it takes so much longer time, due to security checks, sessions processing, streams and all that stuff.*
>
> *The workaround for this particular test case is to keep this stream open, reducing the number of re-negotiation, permission checking, etc.*

Alexey:

> *It reminds me one optimization, that we did for metal pipeline, we have display sync thread which started and stopped on each redraw request.*
>
> *So the optimization was to have a keepalive counter that allows us to keep at alive enought to catch subsequent requests.*
>
> *Probably this approach can be used here.*

Phil:

> *I do not want to overblow this, there are few tests/use cases which really suffers from this, most cases do screen capture a couple of times. So this enhancement could be done later, I don't think that we should rush on that and redone all of that in JDK 21.*
>
> *All of the TCK tests pass, it is a good step that we pass all the conformance tests. It is the some of functional tests failing.*
>
> *The plan is to ship this screencast code in 21, and if somebody happens to run on Wayland it should work modular this crashes. We are not gonna say it is supported, we hoping it should practically for most cases.*

Olivier:

> *XTest will be wired to libei, which will actually move your mouse pointer using the Wayland compositor.*
>
> *The good news is that libei 1.0.0rc1 was released a couple of days ago so we are working hard to try get everything in place, and it is moving nicely.*

Maxim:

> *I fixed a few bugs in our implementation of Wayland toolkit regarding scaling multiscreen environment with different scaling.*
>
> *What I did was that I tried to keep the buffer associated with a window in sync with the scaling of the window of the output where the window is on(use maximum). So if I move window from an output with 200% scale to an output with 100% scale, the buffer actually shrinks, but now I am not sure that I really have to do this, because the window to my does not change at all.*
>
> *Wayland seems to scale the buffer correctly. Do I really need to go through the hassle of adjusting the buffer size and have the buffer scaled to the highest available scale of outputs and leave all work to Wayland?*

Jonas:

> *I would suggest changing the buffer whenever it makes sense. If you don't do this, there are two issues:*
>
> 1. *when the compositor scales down fonts will look slightly more blurry and will be less sharp*
> 2. *If you're working on two monitors (say, 100% and 200%) on a battery-powered machine, and your main program is always on a lower-scaled monitor, all that drawing at 200% and scaling down work will increase power consumption and decrease battery life.*

> *Right now it means that you always check at what outputs you are actually on, an take the maximum scale.*
>
> *There is an extension for the fractional scaling that give you preferred scaling. (Gnome 44+)*

Alexey:

> *We are in progress of implementing Vulkan rendering for pure Wayland prototype*

Alexander:

> *I recall that there was a HiDPI fix for X11 compatibily mode on Wakefield mainling list.*
>
> *Is there any chance that it could be targeted for 21?*

Maxim:

> *It depends on a fix I merged yesterday, so there is no chance that it will go to 21.*

Online Zoom Meeting 9am PDT 27th Apr 2023

Attendees: Victor D'yakov, Alexander Zvegintsev, Olivier Fourdan, Niels de Graef, Phil Race, Maxim Kartashev, Kevin Rushforth

Victor:

> *Alexander, I know you submitted a link to a proposed change to the wakefield sandbox repo, and our plan is to go to open jdk mainline code review as well, right?*
>
> *Please make it much more verbose, as your last submission was really poor on explanation.*

Alexander:

> *Yes, I posted it to the Wakefield sandbox a few weeks ago as a preliminary review of code changes(1, 2) several weeks ago. No work has been done in the last two weeks as we are now doing a test sprint.*
>
> *I will post it to the openjdk repo next week to get it pushed out before the fork of JDK21. It will have more explanations.*

Maxim:

> *Nikita Gubarkov is doing the HiDPI, it relies on me integrating cached bounds and insets for GraphicsDevice into the mainline, and that has not happened yet.*

Phil:

> *Does his code overlap with Alexander's or are they complementary?*

Maxim:

> *I didn't have a chance to check the Alexander's code, but from the descriptions they probably are complementary.*

Phil:

> *We also need to understand the risks of anything that has to do with changes to the X11 compatibility mode, we've had some discussions about how we're going to test all this.*
>
> *We need to be sure that whatever we put in there is not going to break anything else.*
>
> *But the clock is ticking, because RDP1, feature freeze is June 8, so we have 6 weeks.*

Victor:

> *We'll need a CSR to carefully review, it will also require Joe Darcy to jump on it.*

Alexander:

> *There will be a separate PR for changes: code changes and spec changes to make it easier to review(dependent on each other). I'll post a head-up message to wakefield mailing list.*

Niels:

> *About libEI, progress is still being made, and we are getting closer to something stable and ready to merge . One of the latest protcol changes , to not depend on protobuf anymore.*
>
> *We are still planning to make it Gnome 45 and Fedora 39.*

Olivier:

> *For libEI we are expecting last few API tweaks, obviously we want to do all that before we declare the API stable.*

Niels:

> *How is the pure Wayland stuff going on? I understand that it is not going into JDK21*

Maxim:

> *There was no significant progress in the last few months.*

Kevin:

> *The spec changes that are goint into JDK21 should be the same as needed as for the pure Wayland port. There could be some additional things needing to relax.*

Phil:

> *Regarding TCK, there are lots of discussions about how we actually adequately test this from the functional perspective on different distros.*

Alexander:

> *and we still have in issue with Oracle Linux 9.1 that xdg-desktop-portal-gnome is too old, so it keeps asking the user permission for every screenshot.*

Niels:

> *9.2 going to be released in couple of weeks, normally you could try it on CentOS and it should work on RHEL as well.*

Online Zoom Meeting 9am PDT 30th Mar 2023

Attendees: Maxim Kartashev, Alexey Ushakov, Alexander Zvegintsev, Victor D'yakov, Kevin Rushforth, Olivier Fourdan, Jonas Adahl

Alexander:

> *xdg-desktop-portal-gnome package was missing in 9.1, but was present in 9.0.*
>
> *It now comes with 9.1 only with fresh install, system update of an old 9.1 installation didn't work for me, have to install it manually.*
>
> *While the system now successfully reports version 4 of org.freedesktop.portal.ScreenCast. It is minimum required for restore_token functionality.*
>
> *xdg-desktop-portal-gnome package comes outdated for both 9.0 and 9.1(latest available 41.2), to be able make screenshot without confirmation for every attempt we need at least 42+ version..*

> *Added handling of negative scenarios when the user partially or completely denies access to screen data.*
>
> *Now we throw a security exception only if the user denied screen capture at all.*
>
> *If user is allowed to access only some displays, image data of others will be displayed in black, no exceptions or warnings will be thrown.*
>
> *If user wants to change previously made decision, newly added Robot.resetRestoreToken() method can be used.*
>
> *Changed the storage location of restore_token, now instead of a simple file, we use Preferences API.*
>
> *It's still a file, but now it's a standardized XML file, still storing the token in plaintext.*
>
> *I don't think this is a problem though.*
>
> *I also checked probably the most popular product using ScreenCast API - OBS, Open Broadcast Software:*
>
> *They also use store this token in plaintext in their configuration files.*
>
> *I also checked how the prototype behaves with Plasma Desktop Session.*
> *When I get the screen data, everything goes without errors, but the data itself is black.*
> *Apparently, it's because I have an Nvidia graphics card.*
>
> *Another interesting thing is that in response to a call to the Start method, I get a stream that has size, but no position.*
> *This can be a problem, for example when we have two displays with the same resolution.*
> *It is not clear how to distinguish the stream of one screen from the other and thus get the data from the requested display.*

Victor:

> *We have plans to make it, including introduction of header files, in JDK 21 LTS.*
>
> *We will know more on our next meeting in 4 weeks from now.*
>
> *We are usually not making any too risky changes 1-2 weeks before the fork(June 8th), we have only just two months for all this work.*
>
> *It is not our ultimate goal, but a good opportunity.*

Alexey:

> *We have just begun to continue our work on pure Wayland prototype.*
>
> *Nikita Gubarkov has some important changes that could be useful for compatibility mode,*
>
> *he was able to support HiDPI for different monitors using access for underlying wayland subsystem.*
>
> *He said he will be able to port this changes, and probably, to include them into JDK 21 LTS release.*
>
> *It is already integrated in our runtime shipped with our products, looks like all the issues resolved so it is time to move to OpenJDK*

Victor:

> *If this will requires CSR, please do not submit it right before the fork, give it an extra 1-2 weeks.*

Maxim:

> *We could implement some limited support for screenshots for pure wayland.*
>
> *General idea: as long as we can make at least some tests run just a single window, and don't rely on their absolute position,*
>
> *but calculate their offsets just from the top left corner of just one window, we can implement robot color picking, and even screenshoting functionality for this window,*
>
> *by grabbing pixels from associated Wayland buffer.*
>
> *This will work for some tests, especially performance measuring will be more accurate(e.g. color picking at some points frequently).*
>
> *Alexander, can we get recent changes for taking screenshots with ScreenCast API, before they reach JDK 21?*

Alexander:

> *pipewire headers requires legal approval before placing them at openjdk github, but I can make a version to use headers files stored in system and share it.*

Online Zoom Meeting 9am PDT 2nd Mar 2023

Attendees: Alexey Ushakov, Alexander Zvegintsev, Victor D'yakov, Kevin Rushforth, Phil Race, Olivier Fourdan, Jonas Adahl

Alexander :
Pushed the stable version of screencast code to the sand box
Fix for when focus is outside of xserver window
2 places where window focus transfer does not happen and popup remans open
Tried several 3rd party apps and they behave the same so this OK (is it ? - phil)

Working on build changes to build OpenJDK using screencast on OL 7
Have dynamic loading of functions working. Still relying on the pipewire and spa header files.
On RHEL/OL 8 it does not work when try to run screenshot -outdated pipewire
missing xdg portal support.

So probably can't support such older releases as they will never get all the fixes needed.

Last time when trying to take screenshot on RHEL/OL 9 found does not have xdg portal -
Jonas : said xdg portal gnome backend was there in RHEL 9.0 but lost in 9.1
but this was fixed (9.2) ? Alexander would have to check what version he was using
but this is probably the issue

Alexander: we should talk to jtreg folks about not cleaning out various desktop related
variables from the environment that become more important for wakefield

(DBUS_SESSION_BUS_ADDRESS, XDG_SESSION_TYPE and WAYLAND_DISPLAY)
Phil: agreed but our CI system cleans some of these out too, before jtreg even has a chance.
So multiple things to address.

Jonas : how important is it to support restore tokens in persist mode on RHEL9.x/OL9.x
Phil: This is the support to allow permissions to do screen capture to persist across sessions.
It is very much needed for our automated CI testing systems which reboot after every test task.
Would be hard to support a release where it isn't possible.

RHEL bug about backport restore token portal support: https://bugzilla.redhat.com/show_bug.cgi?id=2174946

On Ubuntu 22.04 have pipewire-0.3 and spa-0.2 - are these API stable ?

Jonas (or was it Olivier?) replied pipewire was api stable as of v 0.3.1.X. So probably fine.

Phil:
The build work Alexander is doing is important so that we can build on our CI systems
Otherwise a blocker to integrating even xwayland compatibility support into wayland
We build one binary for many distros and versions .. so there's a LCD requirement.

Alexey :
Talked with Nikita Gubarkov who has been working on hidpi on wayland
(see posts on list at https://mail.openjdk.org/pipermail/wakefield-dev/2023-February/000081.html)
Where should he push this ?
(1) Probably into its own branch at https://github.com/openjdk/wakefield
but
(2) he isn't a project committer so you will need to do it for him

Maxim will hopefully be back soon and will pick up where he left off on wakefield work.

Online Zoom Meeting 9am PDT 2nd Feb 2023

Attendees: Alexey Ushakov, Alexander Zvegintsev, Niels de Graef, Victor D'yakov , Laurent Bourges, Kevin Rushforth, Phil Race, Olivier Fourdan, Jonas Adahl

Alexander:

Updated and slightly redisigned the Known problems and solutions.

The ScreenCast prototype has been redisigned to open/close screencast session during a single call for each screenshot

Some of these changes are already pushed, some will be soon.

It also runs faster than the previous solution.

Alexey:

Did you tried to run some tricky regression tests that require a quick screenshot between some states of controls?

Alexander:

I tried a bunch(they are fine), but haven't run all.

However, some tests that require you to take a screenshot of an icon in the system tray area may fail because an orange"screen is being shared" icon appears while you are taking the screenshot.

Some questions about full support for keyboard/mouse event emulation:

1. What is the status of libEI,
2. The proposed RemoteDesktop solution doesn't provide a concept for retaining user permission between sessions, such as `restore_token` in ScreenCast. When can we expect this to be addressed?
3. Which solution will become usable for our needs first?

Olivier:

There is active work on libEI, its protocol is being actively developed. At current stage it is already works.

Once it finalized and stabilised the follow up work will be to support it on Mutter, xdg-desktop-portal, xdg-desktop-portal-gnome, libportal and libwayland as well.

libEI is planned to ship with Gnome 45(comes with Fedora 39).

Jonas:

xdg-desktop-portal/issues/850 priority can be bumped. At the earliest it can be shipped with Gnome 44(comes with Fedora 38) if it will be fixed soon.

It should be much easier comparing to libEI.

Alexey:

There is a slowdown on pure Wayland port work. More engineers are expected to join this work. Hopefully Maxim will be back this spring.

Phil:

X11 compatibility mode, what are the blockers?

Alexander:

We don't have any.

All of them can be worked around by changing the documentation to reflect the current behavior, modified/excluded tests, even some crashes can be avoided by modifying tests.

Olivier:

xorg/xserver/-/issues/1222 posted a possible fix some time ago, but didn't get any traction

Phil:

All documentation changes should go to the OpenJDK 21 LTS release, even if the work on X11 compatibility mode will not complete by this time. This will allow us to do backports later.

Alexey:

We were able to fix some issues with HiDPI in X11 compatibility mode using some Wayland api besides X11, so probably we will be able to provide our solution.

Online Zoom Meeting 9am PDT 30th June 2022

Attendees: Alexey Ushakov, Olivier Fourdan, Niels de Graef, Jonas Adahl, Kevin Rushforth, Victor Dyakov, Alexander Zvegintsev

Alexander:

Played with gdbus-org.freedesktop.portal.RemoteDesktop API, it looks suitable for mouse/keyboard control for AWT Robot.

Unfortunately it does not support session restoration like gdbus-org.freedesktop.portal.ScreenCast do.

Without it user has to provide "Allow remote interaction" confirmation each time the application is launched..

Is there any plans to add ability to restore session for RemoteDesktop API?

If so, will it come earlier than libEI?

Jonas:

There are such plans, furthermore, RemoteDesktop API will use libEI.

So we can either use RemoteDesktop API or libei directly at some point.

However dates are not clear yet.

Alexey:

A few crashes have been fixed in pure Wayland prototype, and text rendering support has been added.

Software rendering performance is quite good: it is close to 60 FPS(display refresh rate), but it can be worse on a less powerful hardware.

It looks like we can proceed without implementing hardware accelerated pipelines for now.

SwingSet2 works fine, no glitches observed, however input events are not implemented, so there is no way to control the app yet.

We do not expect any several issues with implementing input support.

Online Zoom Meeting 9am PDT 5th May 2022

Attendees: Phil, Kevin, Alexander Zvegintsev, Victor, Zhengyu Gu, Niels de Graef, Maxim Kartashev, Jonas Adahl, Olivier Fourdan, Mario Torres

Alexander : Ubuntu 2204 - no difference from previous releases
Looking at bug where mouse exit event not coming but works fine on native wayland app

Maxim : Nikita has done some work so fonts are being rendered although without hints and AA and sub-pixel but basic rendering working

Also (Maxim) made a preliminary prototype of how we want to start testing pure wayland - custom extension ..
He wasn't able to merge into openjdk repo [Kevin : Maxim missed a step in his project acceptances]

Code available through JB runtime repo
https://github.com/JetBrains/JetBrainsRuntime/tree/wayland-dev/src/java.desktop/share/native/libwakefield

Maxim gave a quick demo of test framework
Lanuch a custom weston instance with Maxim's plugin so you aren't updating the running
desktop so no admin permissions required
The plugin gives access to weston internals -
provides wayland protocols which could be implemented for some other server
https://github.com/mkartashev/wakefield/blob/main/protocol/wakefield.xml

location of surface in abs coords
set location ..
grab portion of screen
get pixel color
would like someone to code review - Java side OK but plugin code is less obvious
Why doesn't work it with weston 10 when it works with weston 9 .. once you get
into accessing internals you really need to build against the exact version of weston you are targetting.

This testing effort is intended to be a stop-gap until we have a proper solution to running tests on
the wayland desktop - not imagined as a solution we can deliver as part of OpenJDK

Online Zoom Meeting 9am PDT 7th April 2022

Attendees: Alexey Ushakov, Olivier Fourdan, Niels de Graef, Jonas Adhal, Sergey Bylokohov,
Alexander Zvegintsev, Zhengyu Gu, Phil Race, Kevin Rushforth, Victor Dyakov

Alexey :
Working on pure wayland, can show AWT and Swing frames with sime simple rendering in paintComponent
Nikita working on font rendering. Will push to branch on openJDK when ready.
3rd thing is AWT robot - Maxim is providing custom extension to Weston instance to get it working.
Alexander : does extension cover screen shot + key press : yes - works even for multimon ...
this is useful for testing but not for production
Sergey : how difficult to install this extension ?

Alexander: Reported a bug about XOR rendering but it could be a driver bug ?
https://gitlab.freedesktop.org/xorg/xserver/-/issues/1333

Screencast still not usable because have to configure it manually every time you start a session
So weston extension looks more promising (for testing)
Need some scripting support
Jonas: maybe not running in a sandbox or something - should be possible to make it remember
what you selected in a dialog.
Maybe be missing a way for portal to associate request with application ID

Screencasting portal configures a token that should make it possible
Screenshot portal doesn't remember that _yet_ though
You can follow this issue for the latter: https://github.com/flatpak/xdg-desktop-portal/issues/649

Niels De Graef : Note that a wl extension protocol that disables security features like that probably won't fly in any compositor

(Jonas & Niels?) : AI: Are there any particualr JDK requirements around input methods - there is a slow moving project
to update in update flawed gtk3 4 protocol .. Phil will have to get back on that after asking around.

Online Zoom Meeting 9am PST 10th Mar 2022

Attendees : Maxim, Olivier Jonas, Alexey, Niels De Graef, Phil, Alexander, Kevin, Victor

Alexander : has been doing testing of current JDK on Oracle Linux 8.5
This uses gnome 3.32 - old gnome shell - so not getting recent fixes
So just a point of reference

Getting some new failures (tests that passed before) on Ubuntu 21.04 and 21.10 - likely this is due to updated OpenGL drivers

Some JDK screen shot tests failing but need to implement something so not an unexpected problem

Olivier : update on Drag and Drop - the upstream fixes have landed in gnome 42 and 40

Niels :
X.org not likely to be removed any time soon by any distro as most are oonly just switching to
wayland as the default. So we have time. Perhaps some cutting edge distro will be first but not soon.

Alexey : Up date on pure wayland work - trying to support just one AWT button but needed to bring in
whole AWT hierarchy - meaning a lot of the equivalent of the Xawt toolkit.

Kind of working but faced with problem of placing location of windows on desktop
can only provide coords for popup surfaces We've talked before about this being part of the design philosophy of wayland
Surely this is needed even for applications that try to place their windows relative to each other ?

Jonas: this has come up in discussions about whether it makes sense ..
GIMP might need this ? Well maybe not GIMP because it is moving away from multiple images but
some medical applications have a need .. but it may be more like session management
1st time launch rearrange your windows. then the extension protocol can be used to save this.
https://gitlab.freedesktop.org/wayland/wayland-protocols/-/merge_requests/18
[Editor : I'm not sure if this would work for JDK needs and how would we expose that protocol to apps ?
It can't just be "java" as that's the platform not the application.]

Alexey is focused on the rendering side. Dmitry Batrak will be looking at event handling.
And Maxim is thinking about Robot - prototyping some new approach to automated testing ... run tests in a weston instance on top of X11 ?
Only useful for testing. Using a custom protocol extension.

Peter Hutterer working on libei and has made a 0.2 release Jonas working on it too.
Concentrating on 2 parts : capturing events and other is sending input events

feature where input captured from compositor - like Synergy which forked into Barrier and then again
That project is to share input devices across multiple computers mouse events go to the other computer when the mouse
goes off edge of screen and emulate events on other computer
API mostly written down but not yet stable.
Peter Hutterer made a 0.2 pre-release.

Blog here :

http://who-t.blogspot.com/2022/03/libei-adding-support-for-passive.html

In the world of sandboxing and flatpak not guaranteed that they have same version of a library.
So not going to make it into any distro even as a pre-release until API and wire format are stable.

Online Zoom Meeting 9am PST 10th Feb 2022

Attendees: Kevin, Victor, Olivier, Mario, Aleksandr, Alexey Ushakov, Maxim Kartashev, Sergey, Zdenek, Zhengyu, Dalibor

Discussion:

We went through all of the open issues on the WIki. Aleksandr will update the bugs with the latest status. The suggestion was made to group them by their status into 2 or three groups (unresolved, fix in progress, resolved). For the unresolved issues, they could be prioritized.

1. JDK-8280982 : java.awt.Robot taking screenshots

Needs to be solved by the Portal team. Not limited to Wayland. This is the number one issue in terms of importance. There is a bug filed against Portal in the flatpak project.

We spent some time discussing the current limitations. What we ultimately need is something that only needs to be configured once by an administrator, and will not produce a popup. Another limitation that needs to be overcome is that the current workarounds only capture the entire screen, which is not sufficient for an API that is often used to read small regions (or a single pixel). This will be too slow. Another idea was discussed to throw a security exception in the screen capture API, but that would require a spec change. It's also not what Java2D does on other platforms (e.g., Mac) when the proper permissions haven't been granted.

2. JDK-8280983 : java.awt.Robot emulating keyboard/mouse events

No new information. Solving this using libEI will be best, and will also solve one of the below issues, but there is no time frame as to when we might expect a fix.

Additionally, using libEI should fix following:
JDK-8280995 : X11 compatibility: Robot.mouseMove does not visually move mouse cursor
JDK-8280990 : X11 compatibility: XTest emulated mouse click does not bring window to front.
JDK-8280988 : X11 compatibility: Click on title to request focus test failures

The last of the above two can be worked around by test changes, but using libEI is a better solution.

3. JDK-8280993 X11 compatibility: Popup menu dismiss in X11 compatibility mode.

This one should be fixable.

4. JDK-8280991 X11 compatibility: fake configure event for XRandR emulation

Should already be fixed in latest version

5. JDK-8280985 X11 compatibility: Wayland crash on huge window

Already fixed

6. JDK-8280987 X11 compatibility: crash when mapping a lot of windows

There are two issues here: one is that we are running out of file descriptors, the other is that this will fill up the Wayland buffers. The first is a simple fix (In progress). The second fix in Wayland itself, not XWayland. It's also in progress, but is a complex fix.

7. JDK-8280992 X11 compatibility: Custom colored cursor might be displayed in black and white

Aleksandr will file a bug if he can come up with a simple reproducer

There are additional issues specific to pure Wayland mode that still need to be filed (this needs more discussion).

Online Zoom Meeting 9am PST 27th Jan 2022

Attendees : Olivier, Phil, Kevin, Aleksandr,, Victor, Niels De Graaf, Alexey Ushakov, Jonas, Sergey, Zdenek, Zhengyu

Aleksandr :
Ubuntu 20.04 - some test problems - tests passed on 20.10 and 21.04

XOR
Memory leak tests - need to increase xMax heap
Mouse enter/exit events
Not always clear if Java problem or Wayland

Let's not spend/waste time on older OSes even LTS versions they'll never have
all the support we need anyway.

Working on migrating bugs / issues just documented on internal wiki to JBS using release==internal

Olivier : DnD also fixed in new OSes
Xwayland is not involved in DnD or copy/paste - standard x atoms

Wayland compositor or applications - x app knows nothing about wayland native so it
is the wayland compositor which does the translation
So doing DnD mutter has to map X11 window and use that to translate
Upstream Gnome 42 fix not yet backported to gnome 41 used by Fedora 35 ..

https://gitlab.gnome.org/GNOME/mutter/-/issues/2042
https://gitlab.gnome.org/GNOME/mutter/-/merge_requests/2259

Can you use latest mutter from main on older OS ? Olivier or Jonas (?) - possible but sounded like a bunch of steps
he will find docs on this
Ed : I think this is the doc meant : https://gitlab.gnome.org/exalm/jhbuild-steps/-/wikis/JHBuild-on-Fedora

Alexey : trying native wayland port rendering using J2D s/w rendering.
Need to fit wayland model to our surface architecture ? Will create a branch in project repo.
Also (AI) will update wiki with info on this

Niels : updated wiki to more clearly describe JDK issues where there are gaps in wayland support

Do AWT Robot docs offer a "way out".
https://docs.oracle.com/en/java/javase/17/docs/api/java.desktop/java/awt/Robot.html

Note that some platforms require special privileges or extensions to access low-level input control. If the current platform configuration does not allow input control, an `AWTException`will be thrown when trying to construct Robot objects. For example, X-Window systems will throw the exception if the XTEST 2.2 standard extension is not supported (or not enabled) by the X server.

Applications that use Robot for purposes other than self-testing should handle these error conditions gracefully.

These words are not intended to say you can be conformant - more about advice that you
may need to update settings add packages to mke your desktop supported.
Analogies such as makung sure you have required X11 packages installed and headful support is available.

Zdenek : Fedora rawhide - dev version of Fedora may be a way of getting VERY latest ahead of a Fedora release

Online Zoom Meeting 9am PST 13th Jan 2022

Attendees : Olivier, Phil, Kevin, Aleksandr, Dalibor, Victor, Niels De Graaf, Alexey Ushakov, Jonas, Mario

he topic of the day was a discussion around Alexey's email to the mailing list ..libei

https://mail.openjdk.java.net/pipermail/wakefield-dev/2021-December/000025.html

proposing what we might do for the "full" wayland native port and the various points it raised

Note: Alexey will add a fleshed out version on the project wiki.

The proposal suggests we'd code to the low level wayland APIs rather than using a high level toolkit (ie GTK4)

As previously observed we should explore both. Probably we could get bootstrapped a lot faster with GTK4
and it would give flexibility for backends so we'd probably not need to create an OGL and Vulkan backend ..
if Vulkan was an option on that distro then we'd expect the underlying Cairo (?) based renderer to be
using it or have it as an option.

Likely there are pros and cons to both options. GTK4 might make assumptions or so things in a way
that isn't compatible with some other functionality. A low-level port might mean a lot of re-inventing
the wheel and having to use functionality that is distor-specific but would have been hidden / abstracted
by the platform GTK 4.

So due diligence on both options, and not expecting completely smooth sailing for either.
Ease of implementation and maintenance and portability are strong arguments, but so are flexibility and
internal consistency in behaviour and behavioural compatibility with the existing X11 implementation.
So if doing it "right" is more work, we'll have to absorb that, and we want a stable base so we aren't
in a perpetual maintenance game.

Mario observed that we need to not get ahead of ourselves because we still have the list
of technical challenges that need to be resolved that apply to X compat mode and to a native port too.

We'll work on those but whilst they work through the eco-system over 2 years (?) we could be
doing something useful ..

One element of the proposal not directly related to wayland is we could merge EDT and Toolkit thread
since we no longer need these separate for applets. Possibly, but we kick off a new thread for modal
dialogs too .. could this be special cased ? Might depend on whether you can limit what happens
in such a case but I'm not sure you can.

We do have these documented on the wiki but it needs another pass to have a consitent format
- what the problem is
- what JDK can do
- what we need from the wayland ecosystem
- how much of a blocker it is
- can we "tweak" the spec - at the cost of making backports harder

We should look at this soon so we can get started on spec tweaks we think may be unavoidable but acceptable ..

Eg What's acceptable for screenshots ? We can't deprecate but can we make something optional ?

Alex Z: Is the SplashScreen API supportable ?

Apparently splash screens display in top lefe (the window positioning problem)

A splash screen (output only) surface role could probably be proposed upstream.
it'd just need a fallback implementation for compositors that doesn't support it, e.g. a dialog

Maybe similar to this PiP request : https://gitlab.freedesktop.org/wayland/wayland-protocols/-/merge_requests/132
For more info on roles: https://wayland-book.com/surfaces/roles.html
Add input to this existing issue : https://gitlab.freedesktop.org/wayland/wayland-protocols/-/issues/67

Accessibility ?

Is there a library ? Like ATK ?
There are existing DBUS based A11Y APIs ..
For the accessibility protocol Jonas is talking about: https://www.freedesktop.org/wiki/Accessibility/AT-SPI2/
For a quick overview of what Emmanuele did for accessibility in GTK4: https://blog.gtk.org/2020/10/21/accessibility-in-gtk-4/

Online Zoom Meeting 9am PST 16th Dec 2021

Attendees : Olivier, Phil, Sergey B., Kevin, Aleksandr, Dalibor, Victor, Niels De Graaf, Alexey Ushakov

Previously discussed upstream DnD bug is fixed upstream - maybe backported to gnome 41 branch ?

On Ubuntu 21.04 using JDK in Xwayland compat. mode DnD from one java app to another (eg) JEdit
gets dropped on native terminal window instead.
DnD is handled by the wayland compositor acting as proxy - Olivier has seen similar issues if you
have wayland native under X11 then it hits the window underneath this should be fixed if you try
Fedora 35 - Ubuntu 21.04 slightly older and presumably doesn't have this fix.

Sergey asked Alexsandr if his DnD testing was using the Robot API or by hand and whether
it is even theoretically possible if wayland doesn't report global screen coords for a window so you
can't move the mouse to a screen position of another windows.
i.e can't move mouse from window 1 -> windows 2 because don't know where window 2 is ...

In other words we have wiggle room in the APIs - and code in the implementation - such that SetBounds calls for a top-level
might be ignored or adjusted, but we rely on correct answers from GetBounds
Olivier : this should actually be OK because in compat mode the apps are talking to an X.org server albeit one acting as a bridge (ie Xwayland)

Some similar discussion where you just want to move the focus from Button A in WIndow 1, to Button B in Window 2
and you need to know the position
In wayland you could set the position by saying loc (x1,y1) in WIndow 2 even if you don't know the real position of Window 2
Not clear (to me) what this would look like for DnD as you want to move the mouse and see it cross the screen outside window boundaries ..
(some of the above needs verifying - outcome of the discussion was not 100% clear to me)
Screencast API is used for automated testing by wayland devs

Would we need to provide new APIs ? And respecify so that existing APIs are "optional".
Could get confusing because other platforms will not have the same issue or model.
Assumes that xwayland compat mode would not need this respecifying to give us backport options.
All too early to know.

Noted that GTK4 was re-specified in a similar way so that things that don't work on wayland are no longer part of the API

How is multi-mon handled in this scenario where you can't position/get position ? Full screen too ?
Full screen is a special case, and you may be able to specify the monitor for a window but that's about it.

Sergey : How does it work for custom decoration on pure wayland
Olivier: Some history here expected client-side decorations was always the answer - so custom decorations would naturally fall from that.
But KDE folks not happy - wanted server side decorations for consistency on the desktop so there is an optional protocol for a wayland
compositor but the client needs to be able to fall back to client-side in case that protocol is not supported on a compositor.

Is there a deadline to ship xwayland compat JDK ?
None yet - still working through these issues - a deadline might be when (say) RHEL shipped without x.org .. which would be a real problem if
we didn't have all the answers ready.

Online Zoom Meeting 9am PST 2nd Dec 2021

Some progress on issues previously seen - some upstream bugs filed and/or fixed.

As reported on the mailing list the DnD issue previously discussed has been fixed upstream

Mapping a single large window : 2 bugs : 1 in wayland 1 in xwayland

Problems with creating a large number of windows is because wayland has fixed buffer sizes for the number of mapped windows

Four, 4K buffers for file descriptors, other requests .. they can fill up
Variable buffer sizes is tricky and a proposed patch is under discussion but not yet accepted.

Some of these problems do not affect running native directly on the hardware with Glimmer and DRM available
but only affect running in a VM where shared memory is used instead.

Online Zoom Meeting 9am PST 18th Nov 2021

We went through some of the issues raised by individuals on the call

Wayland does not provide access to the absolute coordinates of a window on a desktop
or allow specifying it. Whereas X window managers don't make guarantees about positioning but
generally do honour a request and do allow querying. We may have some work to do in making
things work as well as possible with the different approach of wayland
There's (some) discussion here : https://gitlab.freedesktop.org/wayland/wayland-protocols/-/issues/72

There was discussion of ideas such as creating an invisible window that spanned the entire screen
and creating other windows as children of that but this may be fighting another battle where child
windows are assumed to be transient windows.

Focus issues .. and bringing a window to the front or sending it to the back ?
Maybe a way to bring a window to the front with focus but not sending to the back
XDG activation protocol can ask for attention and that usually means being raised to front
https://gitlab.freedesktop.org/wayland/wayland-protocols/-/blob/main/staging/xdg-activation/xdg-activation-v1.xml

Drag and Drop bugs - Alexander having some trouble on Fedora 35

Solutions to some of the challenges are still evolving and different distros may provide different solutions.
There was a discussion and agreement about the importance of implementing to a level that abstracts away
platform-specific code. Coding to something that depends on a particular library that a platform
may not choose to deliver makes our task harder and nearly impossible to deliver one OpenJDK binary
that could run on multiple distros.

All (or at least most) of this is still applicable to both the xwayland case and the wayland case.
Focus for now continues to be on the former. Too early to say if GTK4 or lower-level approach will
ultimately win out for OpenJDK needs.
It was observed that GTK and its print dialog drive you to xprint, which may not be what we want.
Not yet looked into what printing will look like in a native wayland port.

Online Zoom Meeting 9am PDT 16th Sept 2021

Attending : Wakefield committers from Oracle, Redhat, JetBrains and Amazon as well as Wayland developers from Redhat who were invited by Mario.

The intent of this meeting was to have a productive dialog between the OpenJDK developers and Wayland community developers to come to a better shared understanding of the technical challenges and current state of Wayland and identify avenues of investigation.

Some opening remarks were about the Wakefield infrastructure, including this wiki, the Wakefield mailing list and the Wakefield repo.

Now they are up and running we expect to use the mailing list for most communications and document progress etc on this wiki page.

Branches should be created in the Wakefield repo for all work / experiments etc. Don't use the master branch.

In other words, we should use the project infrastructure for everything and publish contributions and ideas there.

The major topic of conversation was around the options for JDK for capturing screenshots and synthesising input events - both for running in X11 compatibility mode or as a native Wayland client

Screen capture is done today by calling a GTK API which is no longer exposed in GTK 4 because core wayland doesn't support it.

We need an alternative, (JDK has a pure X11 fall back based on xwd but we didn't touch on that - it has the same issue)

A few things were thrown out that might merit investigation

https://flatpak.github.io/xdg-desktop-portal/portal-docs.html#gdbus-org.freedesktop.portal.Screenshot

https://docs.flatpak.org/en/latest/portals-gtk.html

https://github.com/flatpak/xdg-desktop-portal#using-portals

https://github.com/flatpak/libportal

The libportal abstraction / helper layer is probably what we'd want.

It is unclear (to me the note taker) if this is readily available on distros alongside wayland - we can't ship it ourselves and it ilikely needs more than a client-side library anyway

JDK for X11 today needs the functionality of the XTEST X11 extension protocol to support the input event side of things.

The wayland desktop answer for this is probably going to be https://gitlab.freedesktop.org/libinput/libei but it is still in development.

It is very unlikely that distros will be ready to ship all the pieces we need in the next 12 months. So the "short term" goal may actually need to wait for somewhat longer than that.

What is the API the clients used for rendering to the wayland client off-screen surface ? GTK defaults to OpenGL (EGL) and falls back to shared memory (ie some software rendering and a copy)

There is also an experimental Vulkan backend.

We use Xrender as the default Linux rendering pipeline today. Maybe we could use OpenGL for the X.org desktop too ? Yes, it is possible that we end up making improvements to the XGL pipeline as a consqeunece of adding EGL support for Wayland so that we could make OpenGL the default there (X.org) too. But it would be a side-benefit not the main goal.

What are people using for window decoration since the wayland model is that the client does the decoration ?I didn't catch all the options here seems there are a few

There was also a question about hidpi causing fuzziness of X11 apps especially for fractional scaling.

The wayland devs confirmed it is a known issue but difficult to solve without making X11 apps far too small to be legible.

The wayland devs input seems to be essential to us navigating a very complex and evolving landscape.

We agreed we need to go off and study some of these APIs and come back with follow up questions when ready.