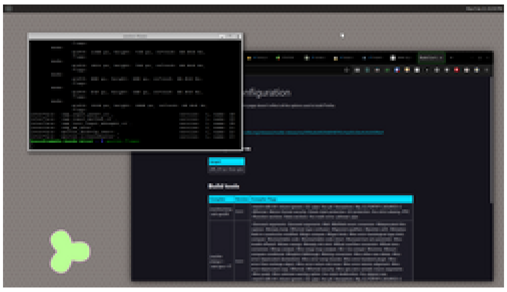
Wikipedia
The Free Encyclopedia

Wayland (protocol)

Wayland is a communication protocol that specifies the communication between a display server and its clients, as well as a C library implementation of that protocol.[9] A display server using the Wayland protocol is called a *Wayland compositor*, because it additionally performs the task of a compositing window manager.

Wayland is developed by a group of volunteers initially led by Kristian Høgsberg as a free and open-source community-driven project with the aim of replacing the X Window System with a secure[10][11][12][13] and simpler windowing system for Linux and other Unix-like operating systems.[9][14] The project's source code is published under the terms of the MIT License, a permissive free software license.[14][6] The Wayland project also develops an implementation of a Wayland compositor called *Weston*.[9]

Wayland



Weston, the reference implementation of a Wayland server

Original author(s)	Kristian Høgsberg
Developer(s)	freedesktop.org et al.
Initial release	30 September 2008[1]
Stable release	Wayland: 1.24,[2] Weston: 14.0.2[3] / 7 July 2025
Repository	gitlab.freedesktop.org/wayland/wayland (https://gitlab.freedesktop.org/wayland/wayland)
Written in	C
Operating system	Official: Linux Unofficial: NetBSD, FreeBSD, OpenBSD, DragonFly BSD[4] Compatibility layer: Haiku[5]
Type	Windowing system Display server
License	MIT License[6][7][8]
Website	wayland.freedesktop.org (https://wayland.freedesktop.org/)

Overview

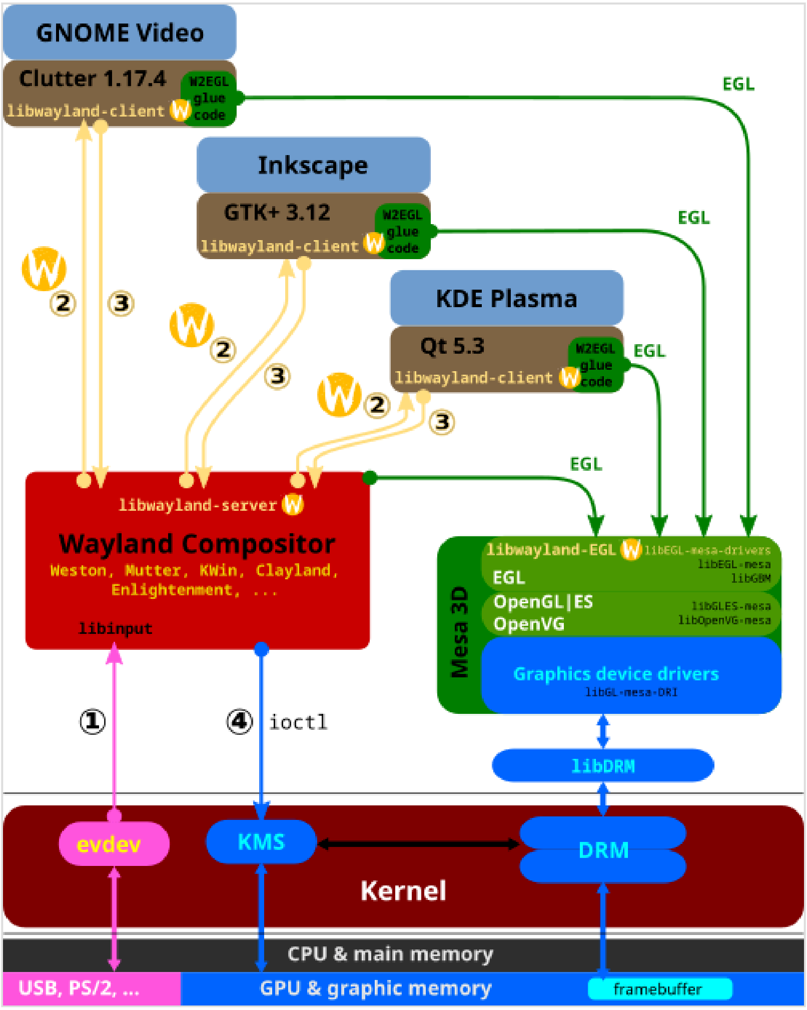
The Wayland Display Server project was started by Red Hat developer Kristian Høgsberg in 2008.[15]

Beginning around 2010, Linux desktop graphics have moved from having "a pile of rendering interfaces... all talking to the X server, which is at the center of the universe" towards putting the Linux kernel and its components (i.e. Direct Rendering Infrastructure (DRI), Direct Rendering Manager (DRM)) "in the middle", with "window systems like X and Wayland ... off in the corner". This will be "a much-simplified graphics system offering more flexibility and better performance".[16]

Høgsberg could have added an extension to X as many recent projects have done, but preferred to "[push] X out of the hotpath between clients and the hardware" for reasons explained in the project's FAQ:[14]

> What's different now is that a lot of infrastructure has moved from the X server into the kernel (memory management, command scheduling, mode setting) or libraries (cairo, pixman, freetype, fontconfig, pango, etc.), and there is very little left that has to happen in a central server process. ... [An X server has] a tremendous amount of functionality that you must support to claim to speak the X protocol, yet nobody will ever use this. ... This includes code tables, glyph rasterization and caching, XLFDs (seriously, XLFDs!), and the entire core rendering API that lets you draw stippled lines, polygons, wide arcs and many more state-of-the-1980s style graphics primitives. For many things we've been able to keep the X.org server modern by adding extension such as XRandR, XRender and COMPOSITE ... With Wayland we can move the X server and all its legacy technology to an optional code path. Getting to a point where the X server is a compatibility option instead of the core rendering system will take a while, but we'll never get there if [we] don't plan for it.

Wayland consists of a protocol and a reference implementation named Weston. The project is also developing



1. The evdev module of the Linux kernel gets an event and sends it to the Wayland compositor.
2. The Wayland compositor looks through its scenegraph to determine which window should receive the event. The scenegraph corresponds to what is on screen and the Wayland compositor understands the transformations that it may have applied to the elements in the scenegraph. Thus, the Wayland compositor can pick the right window and transform the screen coordinates to window local coordinates, by applying the inverse transformations. The types of transformation that can be applied to a window is only restricted to what the compositor can do, as long as it can compute the inverse transformation for the input events.
3. As in the X case, when the client receives the event, it updates the UI in response. But in the Wayland case, the rendering happens by the client via EGL, and the client just sends a request to the compositor to indicate the region that was updated.
4. The Wayland compositor collects damage requests from its clients and then re-composites the screen. The compositor can then directly issue an ioctl to schedule a pageflip with KMS.

versions of GTK and Qt that render to Wayland instead of to X. Most applications are expected to gain support for Wayland through one of these libraries without modification to the application.

Initial versions of Wayland have not provided network transparency, though Høgsberg noted in 2010 that network transparency is possible.[17] It was attempted as a Google Summer of Code project in 2011, but was not successful.[18] Adam Jackson has envisioned providing remote access to a Wayland application by either "pixel-scraping" (like VNC) or getting it to send a "rendering command stream" across the network (as in RDP, SPICE or X11).[19] As of early 2013, Høgsberg was experimenting with network transparency using a proxy Wayland server which sends compressed images to the real compositor.[20][21] In August 2017, GNOME saw the first such pixel-scraping VNC server implementation under Wayland.[22] In modern Wayland compositors, network transparency is handled in an xdg-desktop-portal implementation (https://github.com/flatpak/xdg-desktop-portal) that implements the RemoteDesktop portal.

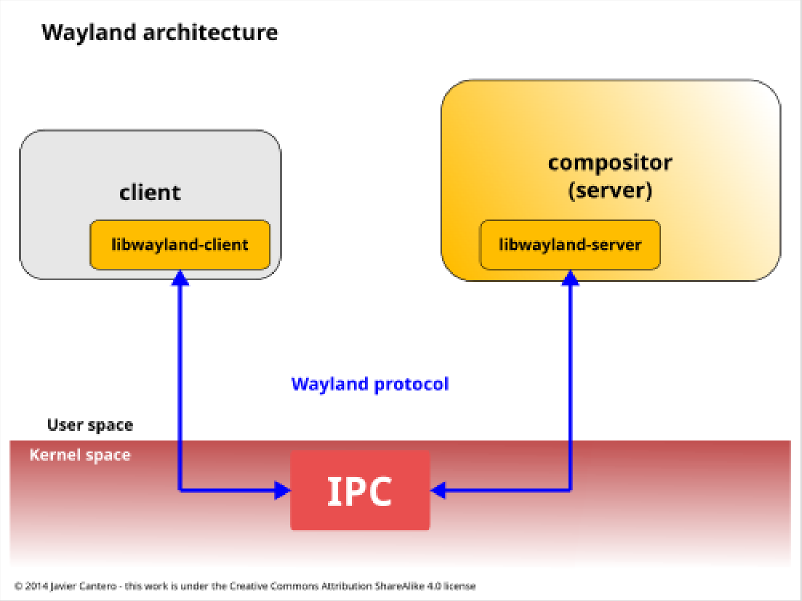
Many Wayland compositors also include an xdg-desktop-portal implementation (https://github.com/flatpak/xdg-desktop-portal) for common tasks such as a native file picker for native applications and sandboxes such as Flatpak (xdg-desktop-portal-gtk (https://github.com/flatpak/xdg-desktop-portal-gtk) is commonly used as a fallback filepicker), screen recording, network transparency, screenshots, color picking, and other tasks that could be seen as needing user intervention and being security risks otherwise. Note that xdg-desktop-portal is not Flatpak or Wayland-specific, and can be used with alternative packaging systems and windowing systems.

Software architecture

Protocol architecture

The Wayland protocol follows a client–server model in which clients are the graphical applications requesting the display of pixel buffers on the screen, and the server (compositor) is the service provider controlling the display of these buffers.

The Wayland reference implementation has been designed as a two-layer protocol:[23]



In the Wayland protocol architecture, a client and a compositor communicate through the Wayland protocol using the reference implementation libraries.

- A low-level layer or *wire protocol* that handles the inter-process communication between the two involved processes—client and compositor—and the marshalling of the data that they interchange. This layer is message-based and usually implemented using the kernel

IPC services, specifically Unix domain sockets in the case of Linux and other Unix-like operating systems.[24]

- A high-level layer built upon it, that handles the information that client and compositor need to exchange to implement the basic features of a window system. This layer is implemented as "an asynchronous object-oriented protocol".[25]

While the low-level layer was written manually in C, the high-level layer is automatically generated from a description of the elements of the protocol stored in XML format.[26] Every time the protocol description of this XML file changes, the C source code that implements such protocol can be regenerated to include the new changes, allowing a very flexible, extensible and error-proof protocol.

The reference implementation of Wayland protocol is split in two libraries: a library to be used by Wayland clients called `libwayland-client` and a library to be used by Wayland compositors called `libwayland-server`.[27]

Protocol overview

The Wayland protocol is described as an "asynchronous object-oriented protocol".[25] *Object-oriented* means that the services offered by the compositor are presented as a series of *objects* living on the same compositor. Each object implements an *interface* which has a name, a number of methods (called *requests*) as well as several associated *events*. Every request and event has zero or more arguments, each one with a name and a data type. The protocol is *asynchronous* in the sense that requests do not have to wait for synchronized replies or ACKs, avoiding round-trip delay time and achieving improved performance.

The Wayland clients can make a request (a method invocation) on some object if the object's interface supports that request. The client must also supply the required data for the arguments of such request. This is the way the clients request services from the compositor. The compositor in turn sends information back to the client by causing the object to emit events (probably with arguments too). These events can be emitted by the compositor as a response to a certain request, or asynchronously, subject to the occurrence of internal events (such as one from an input device) or state changes. The error conditions are also signaled as events by the compositor.[25]

For a client to be able to make a request to an object, it first needs to tell the server the ID number it will use to identify that object.[25] There are two types of objects in the compositor: global objects and non-global objects. Global objects are advertised by the compositor to the clients when they are created (and also when they are destroyed), while non-global objects are usually created by other objects that already exist as part of their functionality.[28]

The interfaces and their requests and events are the core elements that define the Wayland protocol. Each version of the protocol includes a set of interfaces, along with their requests and events, which are expected to be in any Wayland compositor. Optionally, a Wayland compositor may define and implement its own interfaces that support new requests and events, thereby extending functionality beyond the core protocol.[29] To facilitate changes to the protocol, each interface contains a "version number" attribute in addition to its name; this attribute allows for distinguishing variants of the same interface. Each Wayland compositor exposes not only what interfaces are available, but also the supported versions of those interfaces.[30]

Wayland core interfaces

The interfaces of the current version of Wayland protocol are defined in the file `protocol/wayland.xml` of the Wayland source code.[26] This is an XML file that lists the existing interfaces in the current version, along with their requests, events and other attributes. This set of interfaces is the minimum required to be implemented by any Wayland compositor.

Some of the most basic interfaces of the Wayland protocol are:[29]

- *wl_display* – the core global object, a special object to encapsulate the Wayland protocol itself
- *wl_registry* – the global registry object, in which the compositor registers all the global objects that it wants to be available to all clients
- *wl_compositor* – an object that represents the compositor, and is in charge of combining the different surfaces into one output
- *wl_surface* – an object representing a rectangular area on the screen, defined by a location, size and pixel content
- *wl_buffer* – an object that, when attached to a *wl_surface* object, provides its displayable content
- *wl_output* – an object representing the displayable area of a screen
- *wl_pointer*, *wl_keyboard*, *wl_touch* – objects representing different input devices like pointers or keyboards
- *wl_seat* – an object representing a seat (a set of input/output devices) in multiseat configurations

A typical Wayland client session starts by opening a connection to the compositor using the *wl_display* object. This is a special local object that represents the connection and does not live within the server. By using its interface the client can request the *wl_registry* global object from the compositor, where all the global object names live, and bind those that the client is interested in. Usually the client binds at least a *wl_compositor* object from where it will request one or more *wl_surface* objects to show the application output on the display.[28]

Wayland extension interfaces

A Wayland compositor can define and export its own additional interfaces.[29] This feature is used to extend the protocol beyond the basic functionality provided by the core interfaces, and has become the standard way to implement Wayland protocol extensions. Certain compositors can choose to add custom interfaces to provide specialized or unique features. The Wayland reference compositor, Weston, used them to implement new experimental interfaces as a testbed for new concepts and ideas, some of which later became part of the core protocol (such as *wl_subsurface* interface added in Wayland 1.4[31]).

Extension protocols to the core protocol

XDG-Shell protocol

XDG-Shell protocol (see freedesktop.org for XDG) is an extended way to manage surfaces under Wayland compositors (not only Weston). The traditional way to manipulate (maximize, minimize, fullscreen, etc.) surfaces is to use the wl_shell_*() functions, which are part of the core Wayland protocol and live in `libwayland-client`. An implementation of the xdg-shell protocol, on the contrary, is supposed to be provided by the Wayland compositor. So you will find the `xdg-shell-client-protocol.h` header in the Weston source tree.

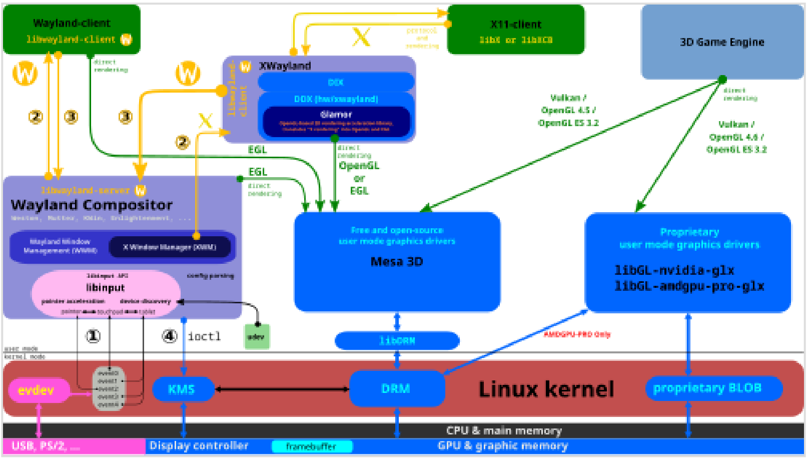
xdg_shell is a protocol aimed to substitute wl_shell in the long term, but will not be part of the Wayland core protocol. It starts as a non-stable API, aimed to be used as a development place at first, and once features are defined as required by several desktop shells, it can be finally made stable. It provides mainly two new interfaces: xdg_surface and xdg_popup. The xdg_surface interface implements a desktop-style window that can be moved, resized, maximized, etc.; it provides a request for creating child/parent relationship. The xdg_popup interface implements a desktop-style popup/menu; an xdg_popup is always transient for another surface, and also has implicit grab.[32]

IVI-Shell protocol

IVI-Shell is an extension to the Wayland core protocol, targeting in-vehicle infotainment (IVI) devices.[33]

Rendering model

The Wayland protocol does not include a rendering API.[34][14][35][36]:2 Instead, Wayland follows a *direct rendering* model, in which the client must render the window contents to a buffer shareable with the compositor.[37] For that purpose, the client can choose to do all the rendering by itself, use a rendering library like Cairo or OpenGL, or rely on the rendering engine of high-level widget libraries with Wayland support, such as Qt or GTK. The client can also optionally use other specialized libraries to perform specific tasks, such as Freetype for font rendering.



Wayland compositor and its clients use EGL to draw directly into the framebuffer; X.Org Server with XWayland and Glamor.

The resulting buffer with the rendered window contents are stored in a *wl_buffer* object. The internal type of this object is implementation dependent. The only requirement is that the content data must be shareable between the client and the compositor. If the client uses a software (CPU) renderer and the result is stored in the system memory, then client and compositor can use shared memory to implement the buffer communication without extra copies. The Wayland protocol already natively provides this kind of shared memory buffer through the *wl_shm*[38] and *wl_shm_pool*[39] interfaces. The drawback of this method is that the compositor may need to do additional work (usually to copy the shared data to the GPU) to display it, which leads to slower graphics performance.

The most typical case is for the client to render directly into a video memory buffer using a hardware (GPU) accelerated API such as OpenGL, OpenGL ES or Vulkan. Client and compositor can share this GPU-space buffer using a special handler to reference it.[40] This method allows the compositor to avoid the extra data copy through itself of the main memory buffer client-to-compositor-to-GPU method, resulting in faster graphics performance, and is therefore the preferred one. The compositor can further optimize the composition of the final scene to be shown on the display by using the same hardware acceleration API as an API client.

When rendering is completed in a shared buffer, the Wayland client should instruct the compositor to present the rendered contents of the buffer on the display. For this purpose, the client binds the buffer object that stores the rendered contents to the surface object, and sends a "commit" request to the surface, transferring the effective control of the buffer to the compositor.[23] Then the client waits for the compositor to release the buffer (signaled by an event) if it wants to reuse the buffer to render another frame, or it can use another buffer to render the new frame, and, when the rendering is finished, bind this new buffer to the surface and commit its contents.[41]:7 The procedure used for rendering, including the number of buffers involved and their management, is entirely under the client control.[41]:7

Comparison with other window systems

Differences between Wayland and X

There are several differences between Wayland and X with regard to performance, code maintainability, and security:[42]

Architecture

The composition manager is a separate, additional feature in X, while Wayland merges display server and compositor as a single function.[43][35] Also, it incorporates some of the tasks of the window manager, which in X is a separate client-side process.[44]

Compositing

Compositing is optional in X, but mandatory in Wayland. Compositing in X is "active"; that is, the compositor must fetch all pixel data, which introduces latency. In Wayland, compositing is "passive", which means the compositor receives pixel data directly from clients.[45]:8–11

Rendering

The X server itself is able to perform rendering, although it can also be instructed to display a rendered window sent by a client. In contrast, Wayland does not expose any API for rendering, but delegates to clients such tasks (including the rendering of fonts, widgets, etc.).[43][35] Window decorations are to be rendered on the client side (e.g., by a graphics toolkit), or on the server side (by the compositor) with the opt-in xdg-decoration (https://wayland.app/protocols/xdg-decoration-unstable-v1) protocol, if the compositor chooses to implement such functionality.[46]

Security

Wayland isolates the input and output of every window, achieving confidentiality, integrity and availability for both. The original X design lacked these important security features,[11][12][13] although some extensions have been developed trying to mitigate it.[47][48][49]

Also, with the vast majority of the code running in the client, less code needs to run with *root* privileges, improving security,[11] although multiple popular Linux distributions now allow the X server to be run without root privileges.[50][51][52][53]

Networking

The X Window System is an architecture that was designed at its core to run over a network. Wayland does not offer network transparency by itself;[14] however, a compositor can implement any remote desktop protocol to achieve remote display. In addition, there is research into Wayland image streaming and compression that would provide remote frame buffer access similar to that of VNC.[21]

Compatibility with X

XWayland is an X Server running as a Wayland client, and thus is capable of displaying native X11 client applications in a Wayland compositor environment.[54] This is similar to the way XQuartz runs X applications in macOS's native windowing system. The goal of XWayland is to facilitate the transition from X Window System to Wayland environments, providing a way to run unported applications in the meantime. XWayland was mainlined into X.Org Server version 1.16.[55]

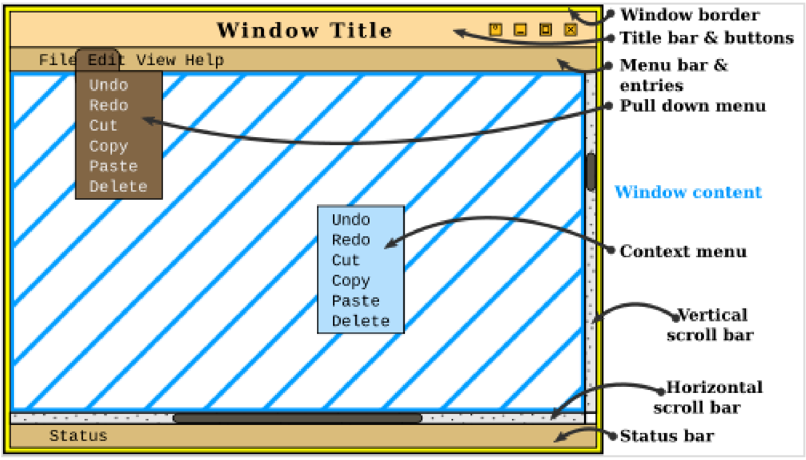
Widget toolkits such as Qt 5 and GTK 3 can switch their graphical back-end at run time,[56] allowing users to choose at load time whether they want to run the application over X or over Wayland. Qt 5 provides the `-platform` command-line option[57] to that effect, whereas GTK 3 lets users select the desired GDK back-end by setting the `GDK_BACKEND` Unix environment variable.[56][58]

Wayland compositors

Display servers that implement the Wayland display server protocol are also called *Wayland compositors* because they additionally perform the task of a compositing window manager.

A library called **wlroots** is a modular Wayland implementation that functions as a base for several compositors[59]

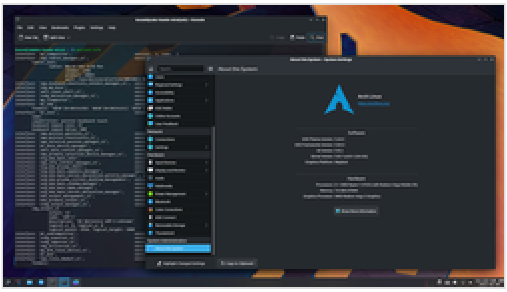
Some notable Wayland compositors are:



Typical elements of a window. The Wayland protocol does not specify what software is responsible for rendering the window decoration.[46]

- Weston – an implementation from the Wayland development team. For details about Weston see below.
- Enlightenment had Wayland support since version 0.20[60]
- KWin, the default wayland compositor of KDE Plasma[61]
- Mutter, the default wayland compositor of GNOME[62]
- Sway – a tiling Wayland compositor, based on wlroots; it is a drop-in replacement for the i3 X11 window manager.[63][64][65]

- Hyprland – a tiling Wayland compositor written in C++. Noteworthy features of Hyprland include dynamic tiling, tabbed windows, and a custom renderer that provides window animations, rounded corners, and Dual-Kawase Blur on transparent windows.[66][67]
- Woodland – wlroots-based window-stacking compositor for Wayland written in C, inspired by TinyWL and focused on simplicity and stability.
- labwc – a wlroots-based window-stacking compositor for Wayland, inspired by Openbox. It follows a similar approach and coding style to Sway.[68]



Plasma 5.24.0 (February 2022) on Wayland (`kwin_wayland` compositor) under Arch Linux

Weston

Weston is a Wayland compositor previously developed as the reference implementation of the protocol by the Wayland project.[69][70] It is written in C and released under the MIT License.

Weston officially supports only Linux due to dependencies on kernel-specific features such as kernel mode-setting (KMS), the Graphics Execution Manager (GEM), and udev. On Linux, it handles input via evdev and buffer management via Generic Buffer Management (GBM). A prototype port for FreeBSD was announced in 2013.[71]

The compositor supports High-bandwidth Digital Content Protection (HDCP)[72] and uses GEM to share buffers between applications and the compositor. It features a plug-in architecture with "shells" that provide elements like docks and panels.[21] Applications are responsible for rendering their own window decorations.

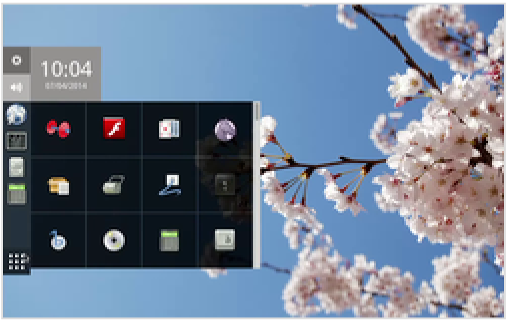
Weston supports rendering via OpenGL ES[73] or the pixman library for software rendering.[74] The full OpenGL stack is avoided to prevent pulling in GLX and other X Window System dependencies.[75]

A remote desktop interface for Weston was proposed in 2013 by a developer from RealVNC.[76]

Maynard

Maynard is a graphical shell and has been written as a plug-in for Weston, just as the GNOME Shell has been written as a plug-in to Mutter.[77]

The Raspberry Pi in collaboration with Collabora released Maynard.[78][79]



Maynard (in January 2017)

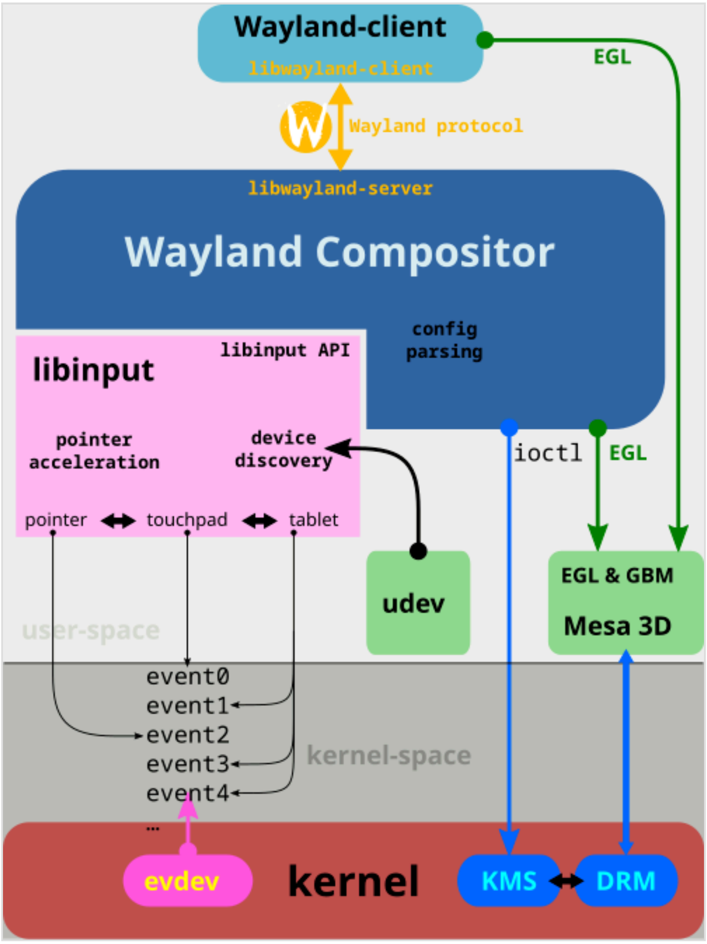
libinput

The Weston code for handling input devices (keyboards, pointers, touch screens, etc.) was split into its own separate library, called *libinput*, for which support was first merged in Weston 1.5.[80][81]

Libinput handles input devices for multiple Wayland compositors and also provides a generic X.Org Server input driver. It aims to provide one implementation for multiple Wayland compositors with a common way to handle input events while minimizing the amount of custom input code compositors need to include. libinput provides device detection (via udev), device handling, input device event processing and abstraction.[82][83]

Version 1.0 of libinput followed version 0.21, and included support for tablets, button sets and touchpad gestures. This version will maintain stable API/ABI.[84]

As GNOME/GTK and KDE Frameworks 5[85] have mainlined the required changes, Fedora 22 will replace X.Org's evdev and Synaptics drivers with libinput.[86]



libinput was created to consolidate the input stack across multiple Wayland compositors.

With version 1.16, the X.Org Server obtained support for the libinput library in form of a wrapper called `xf86-input-libinput`.[87][88]

Wayland Security Module

Wayland Security Module is a proposition that resembles the Linux Security Module interface found in the Linux kernel.[89]

Some applications (especially the ones related to accessibility) require privileged capabilities that should work across different Wayland compositors. Currently, applications under Wayland are generally unable to perform any sensitive tasks such as taking screenshots or injecting input events without going through xdg-desktop-portal (https://github.com/flatpak/xdg-desktop-portal) or obtaining privileged access to the system.

Wayland Security Module is a way to delegate security decisions within the compositor to a centralized security decision engine.[89]

Adoption

The Wayland protocol is designed to be simple so that additional protocols and interfaces need to be defined and implemented to achieve a holistic windowing system. While many graphical toolkits already fully support Wayland, the developers of the graphical shells are cooperating with the Wayland developers to create the necessary additional interfaces.

Desktop Linux distributions

Most major Linux distributions default to using Wayland. Some notable examples are:

- Debian ships Wayland as the default session for GNOME since version 10 (Buster), released 6 July 2019.[90]
- Fedora starting with version 25 (released 22 November 2016) uses Wayland for the default GNOME 3.22 desktop session, with X.Org as a fallback if the graphics driver cannot support Wayland.[91] Fedora uses Wayland as the default for KDE desktop session starting with version 34 (released 27 April 2021)
- Manjaro ships Wayland as default in the Gnome edition of Manjaro 20.2 (Nibia) (released 22 November 2020).[92]
- Raspberry Pi OS, a port of Debian, has offered the option to use Wayland since version 11 (Bullseye), which was released on 3 December 2021. Wayland became the default in version 12 (Bookworm), released on 10 October 2023.
- Red Hat Enterprise Linux ships Wayland as the default session in version 8, released 7 May 2019.[93]
- Ubuntu shipped with Wayland by default in Ubuntu 17.10 (Artful Aardvark).[94] However, Ubuntu 18.04 LTS reverted to X.Org by default due to several issues.[95][96] Since Ubuntu 21.04, Wayland is the default again.[97]
- Slackware Linux included Wayland on 20 February 2020[98] for the development version, -current, which became version 15.0 in 2022. However, Wayland is still not the default.[99]

Toolkit support

Toolkits supporting Wayland include the following:

- EFL has complete Wayland support, except for selection.[100]
- GTK 3.20 has complete Wayland support.[101]
- Qt 5 has complete Wayland support, and can be used to write both Wayland compositors and Wayland clients.
- SDL support for Wayland debuted with the 2.0.2 release[102] and was enabled by default since version 2.0.4.[103]
- GLFW 3.2 has Wayland support.[104]
- FreeGLUT has initial Wayland support.[105]
- FLTK supports Wayland since version 1.4.0 (Nov. 2024).[106]

Desktop environments

Desktop environments that have been ported from X to Wayland include GNOME,[107] KDE Plasma[108] and Enlightenment.[109]

GNOME 3.20 was the first version to have a full Wayland session.[110] GNOME 3.22 included much improved Wayland support across GTK, Mutter, and GNOME Shell.[111] GNOME 3.24 shipped support for the proprietary Nvidia drivers under Wayland.[112]

KDE Plasma started supporting Wayland in version 5.[113] Version 5.4 of Plasma was the first with a full Wayland session.[114] In KDE Plasma 6, Wayland became the default.[115]

In November 2015, Enlightenment e20 was announced with full Wayland support.[116][60][117]

Other software

Other software supporting Wayland includes the following:

- Intelligent Input Bus is working on Wayland support, it could be ready for Fedora 22.[118]
- RealVNC published a Wayland developer preview in July 2014.[76][119][120]
- wayvnc (https://github.com/any1/wayvnc) is a VNC server for wlroots-based Wayland compositors.
- Maliit is an input method framework that runs under Wayland.[121][122][123]
- kmscon supports Wayland with wlterm.[124]
- Mesa has Wayland support integrated.[125]
- Eclipse was made to run on Wayland during a GSoC-Project in 2014.[126]
- The Vulkan WSI (Window System Interface) is a set of API calls that serve a similar purpose as EGL does for OpenGL & OpenGL ES or GLX for OpenGL on X11. Vulkan WSI includes support for Wayland from day one: VK_USE_PLATFORM_WAYLAND_KHR. Vulkan clients can run on unmodified Wayland servers, including Weston, GENIVI LayerManager, Mutter / GNOME Shell, Enlightenment, and more. The WSI allows applications to discover the different GPUs on the system, and display the results of GPU rendering to a window system.[127]
- Waydroid (formerly called Anbox-Halium), a container for Android applications to run on Linux distributions using Wayland.

Mobile and embedded hardware

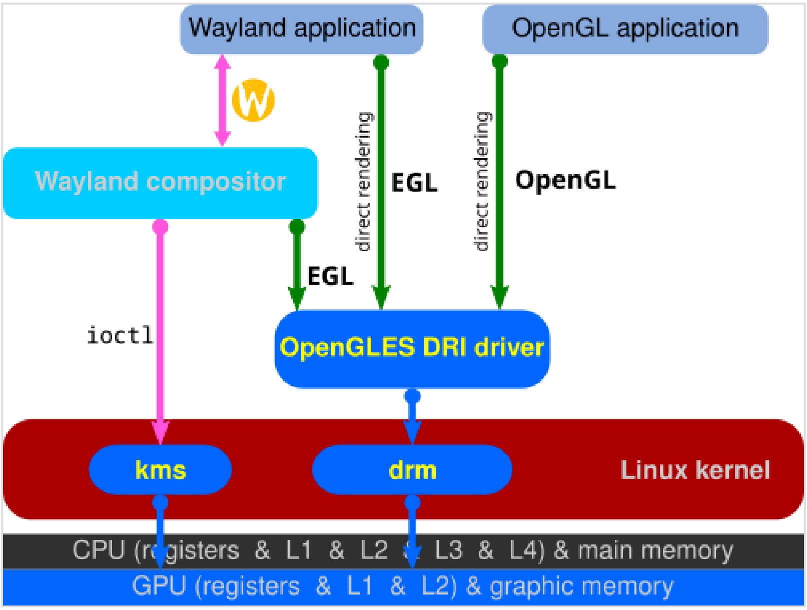
Mobile and embedded hardware supporting Wayland includes the following:

- postmarketOS
- GENIVI Alliance: The GENIVI Aliance, now COVESA, for in-vehicle infotainment (IVI) supports Wayland.[128]
- Jolla: Smartphones from Jolla use Wayland. It is also used as standard when Linux Sailfish OS is used with hardware from other vendors or when it is installed into Android devices by users.[129][130][131]
- Tizen: Starting in version 2.x, Tizen supports Wayland in in-vehicle infotainment (IVI) setups.[132] From 3.0 onward, Tizen defaults to Wayland.[133][134]

Weston running on postmarketOS

History

Kristian Høgsberg, a Linux graphics and X.Org developer who previously worked on AIGLX and DRI2, started Wayland as a spare-time project in 2008 while working for Red Hat.[135][136][137][138] His stated goal was a system in which "every frame is perfect, by which I mean that applications will be able to control the rendering enough that we'll never see tearing, lag, redrawing or flicker." Høgsberg was driving through the town of Wayland, Massachusetts when the underlying concepts "crystallized", hence the name (Weston and Maynard are also nearby towns in the same area, continuing the reference).[137][139]



Wayland uses direct rendering over EGL.

In October 2010, Wayland became a freedesktop.org project.[140][141] As part of the migration, the prior Google Group was replaced by the *wayland-devel* mailing list as the project's central point of discussion and development.

The Wayland client and server libraries were initially released under the MIT License,[142] while the reference compositor Weston and some example clients used the GNU General Public License version 2.[143] Later, all the GPL code was relicensed under the MIT license "to make it easier to move code between the reference implementation and the actual libraries".[144] In 2015 it was discovered that the license text used by Wayland was a slightly different and older version of the MIT license, and the license text was updated to the current version used by the X.Org project (known as MIT Expat License).[6]

Wayland works with all Mesa-compatible drivers with DRI2 support[125] as well as Android drivers via the Hybris project.[145][146][147]

Releases

Major Wayland and Weston releases[148]

Version	Date	Main features		
		Wayland	Weston	Wayland Protocols
0.85	9 February 2012[149]	First release.		
0.95	24 July 2012[150]	Began API stabilization.		
1.0	22 October 2012[151][152]	Stable wayland-client API.		
1.1	15 April 2013[153][154]		Software rendering.[74] FBDEV, RDP backends.	
1.2	12 July 2013[155][156]	Stable wayland-server API.	Color management. Subsurfaces. Raspberry Pi backend.	
1.3	11 October 2013[157]	More pixel formats. Support for language bindings.	Android driver support via libhybris.	
1.4	23 January 2014[31]	New wl_subcompositor and wl_subsurface interfaces.	Multiple framebuffer formats. logind support for rootless Weston.	
1.5	20 May 2014[80]		libinput. Fullscreen shell.	
1.6	19 September 2014[158]		libinput by default.	
1.7	14 February 2015[159][160]		Support for the Wayland presentation extension and for surface roles. IVI shell protocol.	
1.8	2 June 2015[161][162]	Separated headers for core and generated protocol.	Repaint scheduling. Named outputs. Output transformations. Surface-shooting API.	
1.9	21 September 2015[163][164]	Updated license.	Updated license. New test framework. Triple-head DRM compositor. linux_dmabuf extension.	1.0 (2015-11-24)[165]
1.10	17 February 2016[166][167]	Drag-and-drop functionality, grouped pointer events.[168]	Video 4 Linux 2, touch input, debugging improvements.[169]	1.1 (2016-02-16)[170] 1.4 (2016-05-23)[171]
1.11	1 June 2016[172][173]	New backup loading routine, new setup logic.	Proxy wrappers, shared memory changes, Doxygen-generated HTML docs.	1.5 (2016-07-22)[174] 1.7 (2016-08-15)[175]
1.12	21 September 2016[176][177]	Debugging support improved.	libweston and libweston-desktop. Pointer locking and confinement. Relative pointer support.	

1.13	24 February 2017[178]		The ABI of Weston has been changed, thus the new version was named 2.0.0[179] rather than 1.13.0.	1.8 (2017-06-12) 1.10 (2017-07-31)[180]
1.14	8 August 2017[181]		Weston 3.0.0[182] was released at the same time.	1.11 (2017-10-11)[183] 1.13 (2018-02-14)[184]
1.15	9 April 2018[185]		Weston 4.0.0[186] was released at the same time.	1.14 (2018-05-07)[187] 1.16 (2018-07-30)[188]
1.16	24 August 2018[189]		Weston 5.0.0[190] was released at the same time.	1.17 (2018-11-12)[191]
1.17	20 March 2019[192]		Weston 6.0.0[193] was released at the same time.	1.18 (2019-07-25)[194]
1.18	11 February 2020[195]		Weston 7.0.0[196] was released on 2019-08-23. Weston 8.0.0[197] was released on 2020-01-24. Weston 9.0.0[198] was released on 2020-09-04.	1.19 (2020-02-29)[199] 1.20 (2020-02-29)[200]
1.19	27 January 2021[201]			1.21 (2021-04-30)[202] 1.24 (2021-11-23)[203]
1.20	9 December 2021[204]		Weston 10.0.0[205] was released on 2022-02-01. Weston 10.0.5[206] was released on 2023-08-02.	1.25 (2022-01-28)[207]
1.21	30 June 2022[208]		Weston 11.0.0[209] was released on 2022-09-22. Weston 11.0.3[210] was released on 2023-08-02.	1.26 (2022-07-07)[211] 1.31 (2022-11-29)[212]
1.22	4 April 2023[213]		Weston 12.0.0[214] was released on 2023-05-17. Weston 12.0.5[215] was released on 2025-04-29. Weston 13.0.0[216] was released on 2023-11-27. Weston 13.0.4[217] was released on 2025-04-25.	1.32 (2023-07-03)[218] 1.36 (2024-04-26)[219]
1.23	30 May 2024[220]		Weston 14.0.0[221] was released on 2024-09-04. Weston 14.0.2[3] was released on 2025-04-25.	1.37 (2024-08-31)[222] 1.45 (2025-06-13)[223]
1.24	7 July 2025[2]			
1.25				

🟥 Unsupported 🟨 Supported 🟩 **Latest version** 🟪 Future version

See also

Free and open-source software portal

- Mir (software)
- X Window System

References

1. Høgsberg, Kristian (30 September 2008). "Initial commit" (https://gitlab.freedesktop.org/wayland/wayland/-/commit/97f1ebe8d5c2e166fabf757182c289fed266a45a). Retrieved 4 August 2024.
2. "[ANNOUNCE] wayland 1.24.0" (https://lore.freedesktop.org/wayland-devel/plU5smxin5VXbiedTGtL7CNHe0t2mHtgQlQ3-xwREmSSNYaClAcZpEBV8L7JvMT8JHwdD_a_uSFkDUaP45CAWyL545iZh5gInAdToX6hgt4=@emersion.fr/). *Wayland-devel mailing list (lore.freedesktop.org)*. freedesktop.org. 6 July 2025. Retrieved 14 August 2025.
3. "Weston 14.0.2 release" (https://lists.freedesktop.org/archives/wayland-devel/2025-April/044158.html). *Wayland-devel mailing list*. freedesktop.org. April 2025. Retrieved 14 August 2025.
4. "Wayland & Weston Compositor Ported To DragonFlyBSD - Phoronix" (https://www.phoronix.com/scan.php?page=news_item&px=Wayland-DragonFlyBSD-Port). *www.phoronix.com*. Archived (https://web.archive.org/web/20160816072551/https://www.phoronix.com/scan.php?page=news_item&px=Wayland-DragonFlyBSD-Port) from the original on 16 August 2016. Retrieved 20 July 2016.
5. "My progress in Wayland compatibility layer" (https://discuss.haiku-os.org/t/my-progress-in-wayland-compatibility-layer/12373). 24 July 2022.
6. Harrington, Bryce (15 September 2015). "[ANNOUNCE] wayland 1.8.93" (https://lists.freedesktop.org/archives/wayland-devel/2015-September/024226.html). *freedesktop.org* (Mailing list). Archived (https://web.archive.org/web/20201025230202/https://lists.freedesktop.org/archives/wayland-devel/2015-September/024226.html) from the original on 25 October 2020. Retrieved 14 January 2020.
7. "wayland/wayland: root/COPYING" (https://gitlab.freedesktop.org/wayland/wayland/-/blob/main/COPYING). *gitlab.freedesktop.org*. 9 June 2015. Retrieved 4 August 2024.
8. Larabel, Michael (10 June 2015). "Wayland's MIT License To Be Updated/Corrected" (https://www.phoronix.com/scan.php?page=news_item&px=Wayland-License-MIT-Update). *Phoronix.com*. Archived (https://web.archive.org/web/20160425220839/https://www.phoronix.com/scan.php?page=news_item&px=Wayland-License-MIT-Update) from the original on 25 April 2016. Retrieved 17 April 2016.
9. "Wayland" (https://wayland.freedesktop.org/). *Wayland project*. Archived (https://web.archive.org/web/20110302122757/http://wayland.freedesktop.org/) from the original on 2 March 2011. Retrieved 9 May 2016.
10. Sengar, Shivam Singh (16 June 2018). "Wayland v/s Xorg : How Are They Similar & How Are They Different" (https://www.secjuice.com/wayland-vs-xorg/). *secjuice*. Archived (https://web.archive.org/web/20200923224243/https://www.secjuice.com/wayland-vs-xorg/) from the original on 23 September 2020. Retrieved 10 September 2020.
11. Kerrisk, Michael (25 September 2012). "XDC2012: Graphics stack security" (https://lwn.net/Articles/517375/). *LWN.net*. Archived (https://web.archive.org/web/20160328041629/https://lwn.net/Articles/517375/) from the original on 28 March 2016. Retrieved 13 April 2016.

12. Peres, Martin (21 February 2014). "Wayland Compositors - Why and How to Handle Privileged Clients!" (https://mupuf.org/blog/2014/02/19/wayland-compositors-why-and-how-to-handle/). Archived (https://web.archive.org/web/20160310111112/http://www.mupuf.org/blog/2014/02/19/wayland-compositors-why-and-how-to-handle/) from the original on 10 March 2016. Retrieved 17 April 2016.
13. Graesslin, Martin (23 November 2015). "Looking at the security of Plasma/Wayland" (https://blog.martin-graesslin.com/blog/2015/11/looking-at-the-security-of-plasmawayland/). Archived (https://web.archive.org/web/20220827010908/https://blog.martin-graesslin.com/blog/2015/11/looking-at-the-security-of-plasmawayland/) from the original on 27 August 2022. Retrieved 18 April 2016.
14. "Wayland FAQ" (https://wayland.freedesktop.org/faq.html). *Wayland project*. Archived (https://web.archive.org/web/20160402202316/https://wayland.freedesktop.org/faq.html) from the original on 2 April 2016. Retrieved 9 May 2016.
15. Michael Larabel (20 May 2009). "The State Of The Wayland Display Server" (https://www.phoronix.com/scan.php?page=article&item=wayland_q209&num=1). *Phoronix*. Archived (https://web.archive.org/web/20211017212244/https://www.phoronix.com/scan.php?page=article&item=wayland_q209&num=1) from the original on 17 October 2021. Retrieved 17 October 2021.
16. Corbet, Jonathan (5 November 2010). "LPC: Life after X" (https://lwn.net/Articles/413335/). *LWN.net*. Archived (https://web.archive.org/web/20160428103400/https://lwn.net/Articles/413335/) from the original on 28 April 2016. Retrieved 17 April 2016.
17. Høgsberg, Kristian (9 November 2010). "Network transparency argument" (https://lists.freedesktop.org/archives/wayland-devel/2010-November/000097.html). Archived (https://web.archive.org/web/20200815074353/https://lists.freedesktop.org/archives/wayland-devel/2010-November/000097.html) from the original on 15 August 2020. Retrieved 14 January 2020. "Wayland isn't a remote rendering API like X, but that doesn't exclude network transparency. Clients render into a shared buffer and then have to tell the compositor (...) what they changed. The compositor can then send the new pixels in that region out over the network. The Wayland protocol is already violently asynchronous, so it should be able to handle a bit of network lag gracefully. Remote fullscreen video viewing or gaming isn't going to work well, [but] I don't know any other display system that handles that well and transparently."
18. Larabel, Michael (18 August 2011). "Remote Wayland Server Project: Does It Work Yet?" (https://www.phoronix.com/scan.php?page=news_item&px=OTgxMQ). *Phoronix.com*. Archived (https://web.archive.org/web/20160425222517/https://www.phoronix.com/scan.php?page=news_item&px=OTgxMQ) from the original on 25 April 2016. Retrieved 17 April 2016.
19. Jackson, Adam (9 November 2010). "[Re:] Ubuntu moving towards Wayland" (https://web.archive.org/web/20210308134805/https://lists.fedoraproject.org/pipermail/devel/2010-November/145306.html). *devel@lists.fedoraproject.org* (Mailing list). Archived from the original (https://lists.fedoraproject.org/pipermail/devel/2010-November/145306.html) on 8 March 2021. Retrieved 14 January 2020.
20. Stone, Daniel (28 January 2013). *The real story behind Wayland and X* (https://mirror.linux.org.au/pub/linux.conf.au/2013/mp4/The_real_story_behind_Wayland_and_X.mp4) (Speech). linux.conf.au 2013. Canberra. Archived (https://web.archive.org/web/20170810013908/http://mirror.linux.org.au/pub/linux.conf.au/2013/mp4/The_real_story_behind_Wayland_and_X.mp4) from the original on 10 August 2017. Retrieved 13 April 2016.
21. Willis, Nathan (13 February 2013). "LCA: The ways of Wayland" (https://lwn.net/Articles/536862/). *LWN.net*. Archived (https://web.archive.org/web/20160624135326/https://lwn.net/Articles/536862/) from the original on 24 June 2016. Retrieved 19 May 2016.
22. Aleksandersen, Daniel (28 August 2017). "Remote desktop capabilities set to make a comeback in GNOME on Wayland" (https://ctrl.blog/entry/wayland-gnome-remote-desktop). *Ctrl.blog*. Archived (https://web.archive.org/web/20170828230844/https://ctrl.blog/entry/wayland-gnome-remote-desktop) from the original on 28 August 2017. Retrieved 29 August 2017.

23. "The Hello Wayland Tutorial" (https://hdante.wordpress.com/2014/07/08/the-hello-wayland-tutorial/). 8 July 2014. Archived (https://web.archive.org/web/20140713131846/http://hdante.wordpress.com/2014/07/08/the-hello-wayland-tutorial/) from the original on 13 July 2014. Retrieved 25 July 2014.
24. Høgsberg, Kristian. "Chapter 4. Wayland Protocol and Model of Operation" (https://wayland.freedesktop.org/docs/html/ch04.html#sect-Protocol-Wire-Format). *The Wayland Protocol*. Wire Format.
25. Høgsberg, Kristian. "Chapter 4. Wayland Protocol and Model of Operation" (https://wayland.freedesktop.org/docs/html/ch04.html#sect-Protocol-Basic-Principles). *The Wayland Protocol*. Basic Principles.
26. Høgsberg, Kristian (8 May 2024). "protocol/wayland.xml" (https://gitlab.freedesktop.org/wayland/wayland/-/blob/main/protocol/wayland.xml). *gitlab.freedesktop.org*. Retrieved 5 August 2024.
27. Høgsberg, Kristian. "Appendix B. Client API" (https://wayland.freedesktop.org/docs/html/apb.html#id-1.10.2). *The Wayland Protocol*. Introduction.
28. Paalanen, Pekka (25 July 2014). "Wayland protocol design: object lifespan" (https://ppaalanen.blogspot.com/2014/07/wayland-protocol-design-object-lifespan.html). Archived (https://web.archive.org/web/20160529024658/http://ppaalanen.blogspot.com/2014/07/wayland-protocol-design-object-lifespan.html) from the original on 29 May 2016. Retrieved 6 April 2016.
29. Høgsberg, Kristian. "Chapter 4. Wayland Protocol and Model of Operation" (https://wayland.freedesktop.org/docs/html/ch04.html#sect-Protocol-Interfaces). *The Wayland Protocol*. Interfaces.
30. Høgsberg, Kristian. "Chapter 4. Wayland Protocol and Model of Operation" (https://wayland.freedesktop.org/docs/html/ch04.html#sect-Protocol-Versioning). *The Wayland Protocol*. Versioning.
31. Høgsberg, Kristian (24 January 2014). "Wayland and Weston 1.4 is out" (https://lists.freedesktop.org/archives/wayland-devel/2014-January/012903.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20200405181817/https://lists.freedesktop.org/archives/wayland-devel/2014-January/012903.html) from the original on 5 April 2020. Retrieved 14 January 2020.
32. "xdg_shell: Adding a new shell protocol" (https://gitlab.freedesktop.org/wayland/weston/-/commit/3c4dc74cebe1de6f71ec9eeb9794eaaff4327e95). *freedesktop.org*. 3 December 2013. Retrieved 4 August 2024.
33. "GENIVI/wayland-ivi-extension" (https://github.com/GENIVI/wayland-ivi-extension). *GitHub*. 17 November 2021. Archived (https://web.archive.org/web/20170324145817/https://github.com/GENIVI/wayland-ivi-extension) from the original on 24 March 2017. Retrieved 17 August 2016.
34. Høgsberg, Kristian. "Chapter 3. Wayland Architecture" (https://wayland.freedesktop.org/docs/html/ch03.html#sect-Wayland-Architecture-wayland_architecture). *The Wayland Protocol*. X vs. Wayland Architecture.
35. Vervloesem, Koen (15 February 2012). "FOSDEM: The Wayland display server" (https://lwn.net/Articles/481490/). *LWN.net*. Archived (https://web.archive.org/web/20160605025115/https://lwn.net/Articles/481490/) from the original on 5 June 2016. Retrieved 9 May 2016.
36. Barnes, Jesse. "Introduction to Wayland" (https://www.tizen.org/sites/default/files/lfcs2012_barnes.pdf) (PDF). Intel Open Source Technology Center. Archived (https://web.archive.org/web/20160406214636/https://www.tizen.org/sites/default/files/lfcs2012_barnes.pdf) (PDF) from the original on 6 April 2016. Retrieved 18 April 2016. "Does not include a rendering API – Clients use what they want and send buffer handles to the server"
37. Høgsberg, Kristian. "Chapter 3. Wayland Architecture" (https://wayland.freedesktop.org/docs/html/ch03.html#sect-Wayland-Architecture-wayland_rendering). *The Wayland Protocol*. Wayland Rendering.

38. Høgsberg, Kristian. "Appendix A. Wayland Protocol Specification" (https://wayland.freedesktop.org/docs/html/apa.html#protocol-spec-wl_shm). *The Wayland Protocol*. wl_shm - shared memory support.
39. Høgsberg, Kristian. "Appendix A. Wayland Protocol Specification" (https://wayland.freedesktop.org/docs/html/apa.html#protocol-spec-wl_shm_pool). *The Wayland Protocol*. wl_shm_pool - a shared memory pool.
40. Paalanen, Pekka (21 November 2012). "On supporting Wayland GL clients and proprietary embedded platforms" (https://ppaalanen.blogspot.com/2012/11/on-supporting-wayland-gl-clients-and.html). Archived (https://web.archive.org/web/20160406145639/http://ppaalanen.blogspot.com/2012/11/on-supporting-wayland-gl-clients-and.html) from the original on 6 April 2016. Retrieved 6 April 2016. "Buffer sharing works by creating a handle for a buffer, and passing that handle to another process which then uses the handle to make the GPU access again the same buffer."
41. Høgsberg, Kristian. "Wayland Documentation 1.3" (https://wayland.freedesktop.org/docs/pdf/Documentation-1.3-Wayland-en-US.pdf) (PDF). Archived (https://web.archive.org/web/20150910045252/http://wayland.freedesktop.org/docs/pdf/Documentation-1.3-Wayland-en-US.pdf) (PDF) from the original on 10 September 2015. Retrieved 16 March 2016.
42. Griffith, Eric (7 June 2013). "The Wayland Situation: Facts About X vs. Wayland" (https://www.phoronix.com/scan.php?page=article&item=x_wayland_situation). *Phoronix.com*. p. 2. Archived (https://web.archive.org/web/20150924091609/http://www.phoronix.com/scan.php?page=article&item=x_wayland_situation) from the original on 24 September 2015. Retrieved 17 April 2016.
43. "Wayland Architecture" (https://wayland.freedesktop.org/architecture.html). *Wayland project*. Archived (https://web.archive.org/web/20160313044952/https://wayland.freedesktop.org/architecture.html) from the original on 13 March 2016. Retrieved 9 May 2016.
44. Edge, Jake (11 April 2012). "LFCS 2012: X and Wayland" (https://lwn.net/Articles/491509/). *LWN.net*. Archived (https://web.archive.org/web/20160430024705/http://lwn.net/Articles/491509/) from the original on 30 April 2016. Retrieved 17 April 2016.
45. "Wayland/X Compositor Architecture By Example: Enlightenment DR19" (https://events.static.linuxfound.org/sites/events/files/slides/presentation_0.pdf) (PDF). Retrieved 4 August 2024.
46. Graesslin, Martin (7 February 2013). "Client Side Window Decorations and Wayland" (https://blog.martin-graesslin.com/blog/2013/02/client-side-window-decorations-and-wayland/). Archived (https://web.archive.org/web/20160413200359/https://blog.martin-graesslin.com/blog/2013/02/client-side-window-decorations-and-wayland/) from the original on 13 April 2016. Retrieved 17 April 2016.
47. "X.Org Security" (https://www.x.org/wiki/Development/Documentation/Security/). *X.Org Foundation*. Archived (https://web.archive.org/web/20171222000004/https://www.x.org/wiki/Development/Documentation/Security/) from the original on 22 December 2017. Retrieved 2 July 2017. "The X server has long included an extension, SECURITY, which provides support for a simple trusted/untrusted connection model."
48. Wiggins, David P. (15 November 1996). "Security Extension Specification" (https://www.x.org/releases/current/doc/xextproto/security.html). *X Consortium Standard*. Archived (https://web.archive.org/web/20181208112056/https://www.x.org/releases/current/doc/xextproto/security.html) from the original on 8 December 2018. Retrieved 2 July 2017.
49. Walsh, Eamon F. (2009). "X Access Control Extension Specification" (https://www.x.org/releases/X11R7.6/doc/xorg-docs/specs/Xserver/XACE-Spec.html). Archived (https://web.archive.org/web/20160109051310/http://www.x.org/releases/X11R7.6/doc/xorg-docs/specs/Xserver/XACE-Spec.html) from the original on 9 January 2016. Retrieved 2 July 2017.

50. "Debian Moves To Non-Root X.Org Server By Default - Phoronix" (https://www.phoronix.com/scan.php?page=news_item&px=Debian-Non-Root-X). *www.phoronix.com*. Archived (https://web.archive.org/web/20180402225807/https://www.phoronix.com/scan.php?page=news_item&px=Debian-Non-Root-X) from the original on 2 April 2018. Retrieved 2 April 2018.
51. "Non root Xorg - Gentoo Wiki" (https://wiki.gentoo.org/wiki/Non_root_Xorg). *wiki.gentoo.org*. Archived (https://web.archive.org/web/20180402163248/https://wiki.gentoo.org/wiki/Non_root_Xorg) from the original on 2 April 2018. Retrieved 2 April 2018.
52. "X/Rootless - Ubuntu Wiki" (https://wiki.ubuntu.com/X/Rootless). *wiki.ubuntu.com*. Archived (https://web.archive.org/web/20180402225623/https://wiki.ubuntu.com/X/Rootless) from the original on 2 April 2018. Retrieved 2 April 2018.
53. "1078902 – Xorg without root rights" (https://bugzilla.redhat.com/show_bug.cgi?id=1078902). *bugzilla.redhat.com*. Archived (https://web.archive.org/web/20180402225512/https://bugzilla.redhat.com/show_bug.cgi?id=1078902) from the original on 2 April 2018. Retrieved 2 April 2018.
54. "X Clients under Wayland (XWayland)" (https://wayland.freedesktop.org/xserver.html). Wayland project. Archived (https://web.archive.org/web/20220827010905/https://wayland.freedesktop.org/xserver.html) from the original on 27 August 2022. Retrieved 18 July 2014.
55. "ANNOUNCE: xorg-server 1.16.0" (https://lists.x.org/archives/xorg-announce/2014-July/002457.html). freedesktop.org. 17 July 2014. Archived (https://web.archive.org/web/20210308121411/https://lists.x.org/archives/xorg-announce/2014-July/002457.html) from the original on 8 March 2021. Retrieved 14 January 2020.
56. Høgsberg, Kristian (3 January 2011). "Multiple backends for GTK" (https://hoegsberg.blogspot.com/2011/01/multiple-backends-for-gtk.html). Archived (https://web.archive.org/web/20220827010908/https://hoegsberg.blogspot.com/2011/01/multiple-backends-for-gtk.html) from the original on 27 August 2022. Retrieved 10 March 2016.
57. "QtWayland" (https://wiki.qt.io/QtWayland). *Qt Wiki*. Archived (https://web.archive.org/web/20160310090205/http://wiki.qt.io/QtWayland) from the original on 10 March 2016. Retrieved 9 March 2016.
58. "Full Wayland support in GTK+" (https://wiki.gnome.org/Initiatives/Wayland/GTK%2B). *GNOME wiki*. Archived (https://web.archive.org/web/20150907010626/https://wiki.gnome.org/Initiatives/Wayland/GTK%2B) from the original on 7 September 2015. Retrieved 10 March 2016.
59. "README.md" (https://gitlab.freedesktop.org/wlroots/wlroots/-/blob/master/README.md). *wlroots project on GitLab*. 2 February 2023.
60. Larabel, Michael (30 November 2015). "Enlightenment 0.20 Arrives With Full Wayland Support & Better FreeBSD Support" (https://www.phoronix.com/scan.php?page=news_item&px=Enlightenment-0.20-Released). *Phoronix.com*. Archived (https://web.archive.org/web/20160315085313/http://www.phoronix.com/scan.php?page=news_item&px=Enlightenment-0.20-Released) from the original on 15 March 2016. Retrieved 17 April 2016.
61. "Plasma 5.21" (https://kde.org/announcements/plasma/5/5.21.0/). *KDE Community*. 16 February 2021. Archived (https://web.archive.org/web/20210308140934/https://kde.org/announcements/plasma/5/5.21.0/) from the original on 8 March 2021. Retrieved 9 March 2021.
62. "Bump version to 3.13.1" (https://gitlab.gnome.org/GNOME/mutter/-/commit/a7665a3e05bdbfa5540dc1ea92bac10517909d15). 30 April 2014.
63. "Sway" (https://swaywm.org/). *swaywm.org*. Archived (https://web.archive.org/web/20181220230525/https://swaywm.org/) from the original on 20 December 2018. Retrieved 20 December 2018.
64. "swaywm/wlroots" (https://github.com/swaywm/wlroots). *GitHub*. Archived (https://web.archive.org/web/20181219001720/https://github.com/swaywm/wlroots) from the original on 19 December 2018. Retrieved 28 October 2018.

65. "swaywm/sway" (https://github.com/swaywm/sway). *GitHub*. Archived (https://web.archive.org/web/20190828153524/https://github.com/swaywm/sway) from the original on 28 August 2019. Retrieved 28 October 2018.
66. "Hyprland - ArchWiki" (https://wiki.archlinux.org/title/Hyprland). *wiki.archlinux.org*. Retrieved 28 January 2024.
67. "Hyprland: Dynamic tiling window compositor with the looks" (https://hyprland.org/). *hyprland.org*. Retrieved 28 January 2024.
68. "labwc" (https://labwc.github.io/). *labwc.github.io*. Retrieved 28 May 2025.
69. Ser, Simon. "Remove "the reference Wayland compositor" " (https://gitlab.freedesktop.org/wayland/weston/-/merge_requests/1014). *FreeDesktop GitLab*. Retrieved 14 May 2025.
70. "README" (https://gitlab.freedesktop.org/wayland/weston/-/raw/main/README.md). *gitlab.freedesktop.org*. Retrieved 4 August 2024.
71. Larabel, Michael (16 February 2013). "Wayland Begins Porting Process To FreeBSD" (https://www.phoronix.com/scan.php?page=news_item&px=MTMwMzE). *Phoronix.com*. Archived (https://web.archive.org/web/20160425223407/https://www.phoronix.com/scan.php?page=news_item&px=MTMwMzE) from the original on 25 April 2016. Retrieved 17 April 2016.
72. "Adding Content protection support in drm-backend (!48) · Merge Requests · wayland / weston" (https://gitlab.freedesktop.org/wayland/weston/merge_requests/48/diffs?commit_id=05dc6b86431fe040f91ca7b4cf122fa9dc4dd8dd). *GitLab*. 6 November 2018. Archived (https://web.archive.org/web/20220827010905/https://gitlab.freedesktop.org/wayland/weston/-/merge_requests/48/diffs?commit_id=05dc6b86431fe040f91ca7b4cf122fa9dc4dd8dd) from the original on 27 August 2022. Retrieved 27 October 2019.
73. Paalanen, Pekka (10 March 2012). "What does EGL do in the Wayland stack" (https://ppaalanen.blogspot.com/2012/03/what-does-egl-do-in-wayland-stack.html). Archived (https://web.archive.org/web/20161012073727/http://ppaalanen.blogspot.com/2012/03/what-does-egl-do-in-wayland-stack.html) from the original on 12 October 2016. Retrieved 19 May 2016.
74. Larabel, Michael (6 January 2013). "A Software-Based Pixman Renderer For Wayland's Weston" (https://www.phoronix.com/scan.php?page=news_item&px=MTI2Nzg). *Phoronix.com*. Archived (https://web.archive.org/web/20160425225043/https://www.phoronix.com/scan.php?page=news_item&px=MTI2Nzg) from the original on 25 April 2016. Retrieved 17 April 2016.
75. Høgsberg, Kristian (9 December 2010). "Blender3D & cursor clamping" (https://lists.freedesktop.org/archives/wayland-devel/2010-December/000330.html). Archived (https://web.archive.org/web/20201025214539/https://lists.freedesktop.org/archives/wayland-devel/2010-December/000330.html) from the original on 25 October 2020. Retrieved 14 January 2020.
76. "[RFC weston] remote access interface module" (https://lists.freedesktop.org/archives/wayland-devel/2013-October/011536.html). freedesktop.org. 18 October 2013. Archived (https://web.archive.org/web/20201025220521/https://lists.freedesktop.org/archives/wayland-devel/2013-October/011536.html) from the original on 25 October 2020. Retrieved 14 January 2020.
77. "Maynard announcement" (https://blog.barisione.org/2014-04/maynard/). 16 April 2014. Archived (https://web.archive.org/web/20170308063842/http://blog.barisione.org/2014-04/maynard/) from the original on 8 March 2017. Retrieved 16 April 2014.
78. "Raspberry Pi Case Study" (https://www.collabora.com/services/case-studies/raspberrypi/). Collabora. Archived (https://web.archive.org/web/20130901164936/http://www.collabora.com/services/case-studies/raspberrypi/) from the original on 1 September 2013. Retrieved 9 August 2013.
79. "Wayland preview" (https://www.raspberrypi.com/news/wayland-preview/). Raspberry Pi. 24 February 2013. Retrieved 4 August 2024.

80. Høgsberg, Kristian (20 May 2014). "Wayland and Weston 1.5.0 is released" (https://lists.freedesktop.org/archives/wayland-devel/2014-May/014955.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20191019110657/https://lists.freedesktop.org/archives/wayland-devel/2014-May/014955.html) from the original on 19 October 2019. Retrieved 14 January 2020.
81. Ådahl, Jonas (12 November 2013). "[RFC] Common input device library" (https://lists.freedesktop.org/archives/wayland-devel/2013-November/011930.html). Wayland mailing list. Archived (https://web.archive.org/web/20201025230036/https://lists.freedesktop.org/archives/wayland-devel/2013-November/011930.html) from the original on 25 October 2020. Retrieved 14 January 2020.
82. "libinput" (https://www.freedesktop.org/wiki/Software/libinput/). Freedesktop.org. Archived (https://web.archive.org/web/20140419205348/http://www.freedesktop.org/wiki/Software/libinput/) from the original on 19 April 2014. Retrieved 21 May 2014.
83. Hutterer, Peter (8 October 2014). *Consolidating the input stacks with libinput* (https://www.x.org/wiki/Events/XDC2014/XDC2014HuttererLibInput/) (Speech). The X.Org Developer Conference 2014. Bordeaux. Archived (https://web.archive.org/web/20201101021025/https://www.x.org/wiki/Events/XDC2014/XDC2014HuttererLibInput/) from the original on 1 November 2020. Retrieved 14 January 2020.
84. Hutterer, Peter (22 February 2015). "libinput: the road to 1.0" (https://lists.freedesktop.org/archives/wayland-devel/2015-February/020131.html). Archived (https://web.archive.org/web/20201025225206/https://lists.freedesktop.org/archives/wayland-devel/2015-February/020131.html) from the original on 25 October 2020. Retrieved 9 November 2019.
85. "Libinput support added to Touchpad KCM" (https://rajeeshknambiar.wordpress.com/2015/02/22/libinput-support-added-to-touchpad-kcm/). 22 February 2015. Archived (https://web.archive.org/web/20150224161745/https://rajeeshknambiar.wordpress.com/2015/02/22/libinput-support-added-to-touchpad-kcm/) from the original on 24 February 2015. Retrieved 24 February 2015.
86. Goede, Hans de (23 February 2015). "Libinput now enabled as default xorg driver for F-22 workstation installs" (https://web.archive.org/web/20150224160920/https://lists.fedoraproject.org/pipermail/devel/2015-February/208204.html). *devel@lists.fedoraproject.org* (Mailing list). Archived from the original (https://lists.fedoraproject.org/pipermail/devel/2015-February/208204.html) on 24 February 2015. Retrieved 24 February 2015.
87. Hutterer, Peter (24 September 2014). "libinput - a common input stack for Wayland compositors and X.Org drivers" (https://who-t.blogspot.com/2014/09/libinput-common-input-stack-for-wayland.html). Archived (https://web.archive.org/web/20160407130835/http://who-t.blogspot.com/2014/09/libinput-common-input-stack-for-wayland.html) from the original on 7 April 2016. Retrieved 17 April 2016.
88. de Goede, Hans (1 February 2015). "Replacing xorg input - Drivers with libinput" (https://archive.fosdem.org/2015/schedule/event/libinput/attachments/slides/591/export/events/attachments/libinput/slides/591/libinput_xorg.pdf) (PDF). Archived (https://web.archive.org/web/20211122215450/https://archive.fosdem.org/2015/schedule/event/libinput/attachments/slides/591/export/events/attachments/libinput/slides/591/libinput_xorg.pdf) (PDF) from the original on 22 November 2021. Retrieved 18 April 2016.
89. Dodier-Lazaro, Steve; Peres, Martin (9 October 2014). *Security in Wayland-based Desktop Environments: Privileged Clients, Authorization, Authentication and Sandboxing!* (https://phd.mupuf.org/publication/2014/10/09/security-in-wayland-based-desktop-environments/) (Speech). The X.Org Developer Conference 2014. Bordeaux. Archived (https://web.archive.org/web/20160409004644/http://phd.mupuf.org/publication/2014/10/09/security-in-wayland-based-desktop-environments/) from the original on 9 April 2016. Retrieved 18 April 2016.
90. "NewInBuster - Debian Wiki" (https://wiki.debian.org/NewInBuster). *wiki.debian.org*. Archived (https://web.archive.org/web/20190925065917/https://wiki.debian.org/NewInBuster) from the original on 25 September 2019. Retrieved 15 July 2019.

91. "Changes/WaylandByDefault - Fedora Project Wiki" (https://fedoraproject.org/wiki/Changes/WaylandByDefault). *fedoraproject.org*. Archived (https://web.archive.org/web/20151227103524/https://fedoraproject.org/wiki/Changes/WaylandByDefault) from the original on 27 December 2015. Retrieved 9 November 2020.
92. "Manjaro 20.2 Nibia got released" (https://forum.manjaro.org/t/manjaro-20-2-nibia-got-released/41034). 3 December 2020. Archived (https://web.archive.org/web/20201204122710/https://forum.manjaro.org/t/manjaro-20-2-nibia-got-released/41034) from the original on 4 December 2020. Retrieved 3 December 2020.
93. "Release notes for Red Hat Enterprise Linux 8.0" (https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/8.0_release_notes/rhel-8_0_0_release#desktop). *Red Hat Customer Portal*. Archived (https://web.archive.org/web/20201125003517/https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/8.0_release_notes/rhel-8_0_0_release#desktop) from the original on 25 November 2020. Retrieved 25 December 2020.
94. "ReleaseNotes for Ubuntu 17.10" (https://wiki.ubuntu.com/ArtfulAardvark/ReleaseNotes#Ubuntu_Desktop). Canonical. Archived (https://web.archive.org/web/20171124175600/https://wiki.ubuntu.com/ArtfulAardvark/ReleaseNotes#Ubuntu_Desktop) from the original on 24 November 2017. Retrieved 10 November 2017.
95. "Ubuntu 18.04 will revert to long-in-the-tooth Xorg" (https://www.neowin.net/news/ubuntu-1804-will-revert-to-long-in-the-tooth-xorg/). Retrieved 16 December 2023.
96. "Bionic Beaver 18.04 LTS to use Xorg by default" (https://insights.ubuntu.com/2018/01/26/bionic-beaver-18-04-lts-to-use-xorg-by-default/). Canonical. Archived (https://web.archive.org/web/20180218211218/https://insights.ubuntu.com/2018/01/26/bionic-beaver-18-04-lts-to-use-xorg-by-default/) from the original on 18 February 2018. Retrieved 18 February 2018.
97. "Ubuntu 21.04 is here" (https://ubuntu.com/blog/ubuntu-21-04-is-here). *Canonical Ubuntu Blog*. Archived (https://web.archive.org/web/20210422140245/https://ubuntu.com//blog/ubuntu-21-04-is-here) from the original on 22 April 2021. Retrieved 14 June 2021.
98. "Slackware ChangeLogs" (http://www.slackware.com/changelog/current.php?cpu=x86_64). Slackware Linux. Archived (https://web.archive.org/web/20200130153903/http://www.slackware.com/changelog/current.php?cpu=x86_64) from the original on 30 January 2020. Retrieved 26 February 2020.
99. "The Wait is Over! Slackware 15.0 Stable Hit the Streets" (https://linuxiac.com/slackware-15-0-stable/). 4 February 2022. Retrieved 19 June 2025.
100. "Wayland – Enlightenment" (https://web.archive.org/web/20130329104629/http://www.chaosreigns.com/wiki/Wayland_State#The_state_of_Wayland.2C_2013-01-26). Archived from the original (http://www.chaosreigns.com/wiki/Wayland_State#The_state_of_Wayland.2C_2013-01-26) on 29 March 2013. Retrieved 6 March 2013.
101. "GTK+ Roadmap" (https://wiki.gnome.org/Projects/GTK/Roadmap). Retrieved 8 March 2016.
102. Lantinga, Sam (8 March 2014). "SDL 2.0.2 RELEASED!" (https://forums.libsdl.org/viewtopic.php?t=10088). SDL Project. Archived (https://web.archive.org/web/20140315054158/http://forums.libsdl.org/viewtopic.php?t=10088) from the original on 15 March 2014. Retrieved 18 March 2014.
103. Larabel, Michael (9 January 2016). "SDL 2.0.4 Was Quietly Released Last Week With Wayland & Mir By Default" (https://www.phoronix.com/scan.php?page=news_item&px=SDL-2.0.4-Released). *Phoronix.com*. Archived (https://web.archive.org/web/20160316155452/http://www.phoronix.com/scan.php?page=news_item&px=SDL-2.0.4-Released) from the original on 16 March 2016. Retrieved 17 April 2016.

104. Berglund, Camilla (8 April 2014). "Implementation for Wayland · Issue #106 · glfw/glfw · GitHub" (https://github.com/glfw/glfw/issues/106#issuecomment-39872275). *GitHub*. Archived (https://web.archive.org/web/20220827010908/https://github.com/glfw/glfw/issues/106#issuecomment-39872275) from the original on 27 August 2022. Retrieved 14 August 2014.
105. "FreeGLUT: Implement initial Wayland support" (https://github.com/dcnieho/FreeGLUT/commit/9b30564b6d9c9f106c7d079d6cf9207363a49111). *GitHub*. Archived (https://web.archive.org/web/20201110020151/https://github.com/dcnieho/FreeGLUT/commit/9b30564b6d9c9f106c7d079d6cf9207363a49111) from the original on 10 November 2020. Retrieved 7 April 2015.
106. S, A. "FLTK 1.4.0-1 released on Nov. 18, 2024" (https://www.fltk.org/articles.php?L1955). *FLTK*. AlbrechtS.
107. "GNOME Initiatives - Wayland" (https://wiki.gnome.org/Initiatives/Wayland). *GNOME Wiki*. Archived (https://web.archive.org/web/20160307164016/https://wiki.gnome.org/Initiatives/Wayland) from the original on 7 March 2016. Retrieved 8 March 2016.
108. "KWin/Wayland" (https://community.kde.org/KWin/Wayland). *KDE Community Wiki*. Archived (https://web.archive.org/web/20160310115021/https://community.kde.org/KWin/Wayland) from the original on 10 March 2016. Retrieved 8 March 2016.
109. "Enlightenment - Wayland" (https://www.enlightenment.org/about-wayland). *Enlightenment.org*. Archived (https://web.archive.org/web/20160309102604/https://www.enlightenment.org/about-wayland) from the original on 9 March 2016. Retrieved 8 March 2016.
110. "ReleasePlanning/FeaturesPlans" (https://wiki.gnome.org/ReleasePlanning/FeaturePlans). GNOME Project. Archived (https://web.archive.org/web/20150531015800/https://wiki.gnome.org/ReleasePlanning/FeaturePlans) from the original on 31 May 2015. Retrieved 31 May 2015.
111. "A Look At The Exciting Features/Improvements Of GNOME 3.22" (https://www.phoronix.com/scan.php?page=article&item=gnome-322-features&num=1). Phoronix. Archived (https://web.archive.org/web/20200812095126/https://www.phoronix.com/scan.php?page=article&item=gnome-322-features&num=1) from the original on 12 August 2020. Retrieved 14 January 2020.
112. "GNOME Lands Mainline NVIDIA Wayland Support Using EGLStreams" (https://www.phoronix.com/scan.php?page=news_item&px=GNOME-Mutter-Mainline-EGLStream). Phoronix. Archived (https://web.archive.org/web/20190808131309/https://www.phoronix.com/scan.php?page=news_item&px=GNOME-Mutter-Mainline-EGLStream) from the original on 8 August 2019. Retrieved 14 January 2020.
113. "Plasma's Road to Wayland" (https://web.archive.org/web/20140727191427/http://vizzzion.org/blog/2014/07/plasmas-road-to-wayland/). 25 July 2014. Archived from the original (http://vizzzion.org/blog/2014/07/plasmas-road-to-wayland/) on 27 July 2014. Retrieved 25 July 2014.
114. Graesslin, Martin (29 June 2015). "Four years later" (https://blog.martin-graesslin.com/blog/2015/06/four-years-later/). Archived (https://web.archive.org/web/20160331202154/http://blog.martin-graesslin.com/blog/2015/06/four-years-later/) from the original on 31 March 2016. Retrieved 17 April 2016.
115. Wallen, Jack (14 February 2024). "The first Linux distribution to deliver a pure KDE Plasma 6 environment is here" (https://www.zdnet.com/article/the-first-linux-distribution-to-deliver-a-pure-kde-plasma-6-environment-is-here/). *ZDNET*. Retrieved 24 March 2024.
116. "Enlightenment DR 0.20.0 Release" (https://web.archive.org/web/20180621015953/https://phab.enlightenment.org/phame/post/view/167/enlightenment_dr_0.20.0_release/). *Enlightenment.org*. Archived from the original (https://phab.enlightenment.org/phame/post/view/167/enlightenment_dr_0.20.0_release/) on 21 June 2018. Retrieved 8 March 2016.

117. "The Enlightenment of Wayland" (https://fosdem.org/2016/schedule/event/enlightenment_of_wayland/). *FOSDEM.org*. Archived (https://web.archive.org/web/20160309103352/https://fosdem.org/2016/schedule/event/enlightenment_of_wayland/) from the original on 9 March 2016. Retrieved 8 March 2016.
118. Schaller, Christian (3 July 2014). "Wayland in Fedora Update" (https://blogs.gnome.org/uraeus/2014/07/03/wayland-in-fedora-update/). *blogs.gnome.org*. Archived (https://web.archive.org/web/20160501170950/https://blogs.gnome.org/uraeus/2014/07/03/wayland-in-fedora-update/) from the original on 1 May 2016. Retrieved 17 April 2016.
119. "VNC® Wayland Developer Preview" (https://web.archive.org/web/20140714205900/http://www.realvnc.com/products/vnc/wayland/). 8 July 2014. Archived from the original (https://www.realvnc.com/products/vnc/wayland/) on 14 July 2014.
120. "RealVNC Wayland developer preview email" (https://lists.freedesktop.org/archives/wayland-devel/2014-July/015949.html). freedesktop.org. 9 July 2014. Archived (https://web.archive.org/web/20201025220610/https://lists.freedesktop.org/archives/wayland-devel/2014-July/015949.html) from the original on 25 October 2020. Retrieved 14 January 2020.
121. "Maliit Status Update" (https://web.archive.org/web/20130517213318/http://mikhas.posterous.com/maliit-status-update). Posterous. 2 April 2013. Archived from the original (http://mikhas.posterous.com/maliit-status-update) on 17 May 2013. Retrieved 14 October 2013.
122. "More Maliit Keyboard Improvements: QtQuick2" (https://www.murrayc.com/permalink/2013/04/02/more-maliit-keyboard-improvements-qtquick2/). Murray's Blog. 2 April 2013. Archived (https://web.archive.org/web/20131013194426/http://www.murrayc.com/permalink/2013/04/02/more-maliit-keyboard-improvements-qtquick2/) from the original on 13 October 2013. Retrieved 14 October 2013.
123. "Maliit under Wayland" (https://web.archive.org/web/20130611231511/https://wiki.maliit.org/Wayland_Input_Method_System_Proposal). Archived from the original (https://wiki.maliit.org/Wayland_Input_Method_System_Proposal) on 11 June 2013. Retrieved 14 September 2013.
124. "wlterm" (https://www.freedesktop.org/wiki/Software/kmscon/wlterm/). Freedesktop.org. Archived (https://web.archive.org/web/20140714182643/http://www.freedesktop.org/wiki/Software/kmscon/wlterm/) from the original on 14 July 2014. Retrieved 8 July 2014.
125. Hillesley, p. 3.
126. "Eclipse now runs on Wayland" (http://blog.vogella.com/2014/08/18/eclipse-now-runs-on-wayland/). 18 August 2014. Archived (https://web.archive.org/web/20140823203722/http://blog.vogella.com/2014/08/18/eclipse-now-runs-on-wayland/) from the original on 23 August 2014. Retrieved 17 September 2014.
127. Stone, Daniel (16 February 2016). "Vulkan 1.0 specification released with day-one support for Wayland" (https://www.collabora.com/about-us/blog/2016/02/16/vulkan-1.0-specification-released-with-day-one-support-for-wayland/). Archived (https://web.archive.org/web/20160624004927/https://www.collabora.com/about-us/blog/2016/02/16/vulkan-1.0-specification-released-with-day-one-support-for-wayland/) from the original on 24 June 2016. Retrieved 27 February 2016.
128. "Wayland Backend DRM | IVI Layer Management" (https://web.archive.org/web/20140201161624/http://projects.genivi.org/ivi-layer-management/node/17). GENIVI Alliance. Archived from the original (http://projects.genivi.org/ivi-layer-management/node/17) on 1 February 2014. Retrieved 15 July 2013.
129. "The First Jolla Smartphone Runs With Wayland" (http://linuxg.net/the-first-jolla-smartphone-runs-with-wayland/). LinuxG.net. 14 July 2013. Archived (https://web.archive.org/web/20140628095653/http://linuxg.net/the-first-jolla-smartphone-runs-with-wayland/) from the original on 28 June 2014. Retrieved 8 October 2013.
130. VDVsx [@VDVsx] (13 July 2013). "#sailfishos main components diagram. #Qt5 #Wayland #JollaHQ #Akademy" (https://x.com/VDVsx/status/356029626759385088) (Tweet) – via Twitter.

131. Jolla [@JollaHQ] (13 July 2013). "@faenil @PeppeLaKappa @VDVsx our first Jolla will ship with wayland, yes" (https://x.com/JollaHQ/status/356034168351756290) (Tweet) – via Twitter.
132. "IVI/IVI Setup" (https://wiki.tizen.org/w/index.php?title=IVI%2FIVI_Setup&oldid=1037). Tizen Wiki. Archived (https://web.archive.org/web/20170210051856/https://wiki.tizen.org/w/index.php?title=IVI%2FIVI_Setup&oldid=1037) from the original on 10 February 2017. Retrieved 8 April 2013.
133. VanCutsem, Geoffroy (10 July 2013). "[IVI] Tizen IVI 3.0-M1 released" (https://lists.tizen.org/pipermail/ivi/2013-July/000563.html). *IVI* (Mailing list). Archived (https://web.archive.org/web/20131002140438/https://lists.tizen.org/pipermail/ivi/2013-July/000563.html) from the original on 2 October 2013. Retrieved 13 July 2013.
134. Amadeo, Ron (12 May 2017). "The Samsung Z4 is Tizen's new flagship smartphone" (https://arstechnica.com/gadgets/2017/05/the-samsung-z4-is-tizens-new-flagship-smartphone/). *Ars Technica*. Archived (https://web.archive.org/web/20170513000659/https://arstechnica.com/gadgets/2017/05/the-samsung-z4-is-tizens-new-flagship-smartphone/) from the original on 13 May 2017. Retrieved 13 May 2017.
135. Høgsberg, Kristian (3 November 2008). "Premature publicity is better than no publicity" (https://hoegsberg.blogspot.com/2008/11/premature-publicity-is-better-than-no.html). Archived (https://web.archive.org/web/20160328164015/http://hoegsberg.blogspot.com/2008/11/premature-publicity-is-better-than-no.html) from the original on 28 March 2016. Retrieved 6 April 2016.
136. "Interview: Kristian Høgsberg" (https://archive.fosdem.org/2012/interview/kristian-hogsberg.html). *FOSDEM Archive*. 29 January 2012. Archived (https://web.archive.org/web/20160304040036/https://archive.fosdem.org/2012/interview/kristian-hogsberg.html) from the original on 4 March 2016. Retrieved 8 March 2016.
137. Hillesley, Richard (13 February 2012). "Wayland - Beyond X" (https://web.archive.org/web/20131206201020/http://www.h-online.com/open/features/Wayland-Beyond-X-1432046.html). *The H Open*. Heise Media UK. Archived from the original (http://www.h-online.com/open/features/Wayland-Beyond-X-1432046.html) on 6 December 2013. Retrieved 8 March 2016.
138. Høgsberg, Kristian. "Wayland – A New Display Server for Linux" (https://web.archive.org/web/20170811103724/http://linuxplumbersconf.org/ocw/users/73). Linux Plumbers Conference, 2009. Archived from the original (https://linuxplumbersconf.org/ocw/users/73) on 11 August 2017.
139. Jenkins, Evan (22 March 2011). "The Linux graphics stack from X to Wayland" (https://arstechnica.com/information-technology/2011/03/the-linux-graphics-stack-from-x-to-wayland/). *Ars Technica*. Archived (https://web.archive.org/web/20160410131105/http://arstechnica.com/information-technology/2011/03/the-linux-graphics-stack-from-x-to-wayland/) from the original on 10 April 2016. Retrieved 17 April 2016.
140. Larabel, Michael (29 October 2010). "Wayland Becomes A FreeDesktop.org Project" (https://www.phoronix.com/scan.php?page=news_item&px=ODczNQ). *Phoronix.com*. Archived (https://web.archive.org/web/20160414125119/http://www.phoronix.com/scan.php?page=news_item&px=ODczNQ) from the original on 14 April 2016. Retrieved 17 April 2016.
141. Høgsberg, Kristian (29 October 2010). "Moving to freedesktop.org" (https://groups.google.com/forum/#!topic/wayland-display-server/sfvLwLaHwsE). Archived (https://web.archive.org/web/20121109222635/http://groups.google.com/group/rec.arts.comics.misc/browse_frm/thread/758fa9d13d234863/8d4e36d21137d39e?lnk=st&q=%22Archie+goes+unchallenged%22#!topic/wayland-display-server/sfvLwLaHwsE) from the original on 9 November 2012. Retrieved 31 July 2013.

142. Høgsberg, Kristian (3 December 2008). "Wayland is now under MIT license" (https://groups.google.com/forum/#!topic/wayland-display-server/daWzNkr5erE). *wayland-display-server* (Mailing list). Archived (https://web.archive.org/web/20121109222635/http://groups.google.com/group/rec.arts.comics.misc/browse_frm/thread/758fa9d13d234863/8d4e36d21137d39e?lnk=st&q=%22Archie+goes+unchallenged%22#!topic/wayland-display-server/daWzNkr5erE) from the original on 9 November 2012. Retrieved 8 March 2016.
143. Høgsberg, Kristian (22 November 2010). "Wayland license clarification" (https://lists.freedesktop.org/archives/wayland-devel/2010-November/000249.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20160309101222/https://lists.freedesktop.org/archives/wayland-devel/2010-November/000249.html) from the original on 9 March 2016. Retrieved 8 March 2016.
144. Høgsberg, Kristian (19 September 2011). "License update" (https://lists.freedesktop.org/archives/wayland-devel/2011-September/001386.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20160429042115/https://lists.freedesktop.org/archives/wayland-devel/2011-September/001386.html) from the original on 29 April 2016. Retrieved 8 March 2016.
145. Munk, Carsten (11 April 2013). "Wayland utilizing Android GPU drivers on glibc based systems, Part 1" (https://mer-project.blogspot.com/2013/04/wayland-utilizing-android-gpu-drivers.html). Mer Project. Archived (https://web.archive.org/web/20130419050731/http://mer-project.blogspot.com/2013/04/wayland-utilizing-android-gpu-drivers.html) from the original on 19 April 2013. Retrieved 3 July 2013.
146. Munk, Carsten (8 June 2013). "Wayland utilizing Android GPU drivers on glibc based systems, Part 2" (https://mer-project.blogspot.com/2013/05/wayland-utilizing-android-gpu-drivers.html). Mer Project. Archived (https://web.archive.org/web/20131024095141/http://mer-project.blogspot.com/2013/05/wayland-utilizing-android-gpu-drivers.html) from the original on 24 October 2013. Retrieved 3 July 2013.
147. Larabel, Michael (11 April 2013). "Jolla Brings Wayland Atop Android GPU Drivers" (https://www.phoronix.com/scan.php?page=news_item&px=MTM0OTE). *Phoronix.com*. Archived (https://web.archive.org/web/20160425233250/https://www.phoronix.com/scan.php?page=news_item&px=MTM0OTE) from the original on 25 April 2016. Retrieved 17 April 2016.
148. "Wayland" (https://wayland.freedesktop.org/releases.html). Wayland.freedesktop.org. Archived (https://web.archive.org/web/20130707081706/http://wayland.freedesktop.org/releases.html) from the original on 7 July 2013. Retrieved 15 July 2013.
149. Høgsberg, Kristian (9 February 2012). "[ANNOUNCE] Wayland and Weston 0.85.0 released" (https://lists.freedesktop.org/archives/wayland-devel/2012-February/002072.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20200114134921/https://lists.freedesktop.org/archives/wayland-devel/2012-February/002072.html) from the original on 14 January 2020. Retrieved 14 January 2020.
150. Høgsberg, Kristian (24 July 2012). "Wayland and Weston 0.95.0 released" (https://lists.freedesktop.org/archives/wayland-devel/2012-July/004542.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20201025215619/https://lists.freedesktop.org/archives/wayland-devel/2012-July/004542.html) from the original on 25 October 2020. Retrieved 14 January 2020.
151. Høgsberg, Kristian (22 October 2012). "Wayland and Weston 1.0" (https://lists.freedesktop.org/archives/wayland-devel/2012-October/005967.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20190823084549/https://lists.freedesktop.org/archives/wayland-devel/2012-October/005967.html) from the original on 23 August 2019. Retrieved 14 January 2020.
152. Scherschel, Fabian (23 October 2012). "Wayland's 1.0 milestone fixes graphics protocol" (https://web.archive.org/web/20131207164236/http://www.h-online.com/open/news/item/Wayland-s-1-0-milestone-fixes-graphics-protocol-1734638.html). *The H Open*. Heise Media UK. Archived from the original (http://www.h-online.com/open/news/item/Wayland-s-1-0-milestone-fixes-graphics-protocol-1734638.html) on 7 December 2013. Retrieved 17 April 2016.

153. Larabel, Michael (16 April 2013). "Wayland 1.1 Officially Released With Weston 1.1" (https://www.phoronix.com/scan.php?page=news_item&px=MTM1Mjk). *Phoronix.com*. Archived (https://web.archive.org/web/20160425213252/https://www.phoronix.com/scan.php?page=news_item&px=MTM1Mjk) from the original on 25 April 2016. Retrieved 17 April 2016.
154. Høgsberg, Kristian (15 April 2013). "1.1 Released" (https://lists.freedesktop.org/archives/wayland-devel/2013-April/008631.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20190527164726/https://lists.freedesktop.org/archives/wayland-devel/2013-April/008631.html) from the original on 27 May 2019. Retrieved 14 January 2020.
155. Larabel, Michael (13 July 2013). "Wayland 1.2.0 Released, Joined By Weston Compositor" (https://www.phoronix.com/scan.php?page=news_item&px=MTQwOTg). *Phoronix.com*. Archived (https://web.archive.org/web/20160414124843/http://www.phoronix.com/scan.php?page=news_item&px=MTQwOTg) from the original on 14 April 2016. Retrieved 17 April 2016.
156. Høgsberg, Kristian (12 July 2013). "Wayland and Weston 1.2.0 released" (https://lists.freedesktop.org/archives/wayland-devel/2013-July/010278.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20190625192111/https://lists.freedesktop.org/archives/wayland-devel/2013-July/010278.html) from the original on 25 June 2019. Retrieved 14 January 2020.
157. Høgsberg, Kristian (11 October 2013). "Wayland and Weston 1.3 releases are out" (https://lists.freedesktop.org/archives/wayland-devel/2013-October/011419.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20200808090510/https://lists.freedesktop.org/archives/wayland-devel/2013-October/011419.html) from the original on 8 August 2020. Retrieved 11 April 2020.
158. Paalanen, Pekka (19 September 2014). "Wayland and Weston 1.6.0 released" (https://lists.freedesktop.org/archives/wayland-devel/2014-September/017396.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20190527164732/https://lists.freedesktop.org/archives/wayland-devel/2014-September/017396.html) from the original on 27 May 2019. Retrieved 14 January 2020.
159. Harrington, Bryce (14 February 2015). "[ANNOUNCE] wayland 1.7.0" (https://lists.freedesktop.org/archives/wayland-devel/2015-February/019976.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20200405181859/https://lists.freedesktop.org/archives/wayland-devel/2015-February/019976.html) from the original on 5 April 2020. Retrieved 14 January 2020.
160. Harrington, Bryce (14 February 2015). "[ANNOUNCE] weston 1.7.0" (https://lists.freedesktop.org/archives/wayland-devel/2015-February/019977.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20200405181859/https://lists.freedesktop.org/archives/wayland-devel/2015-February/019977.html) from the original on 5 April 2020. Retrieved 14 January 2020.
161. Harrington, Bryce (2 June 2015). "[ANNOUNCE] wayland 1.8.0" (https://lists.freedesktop.org/archives/wayland-devel/2015-June/022415.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20200815091351/https://lists.freedesktop.org/archives/wayland-devel/2015-June/022415.html) from the original on 15 August 2020. Retrieved 14 January 2020.
162. Harrington, Bryce (2 June 2015). "[ANNOUNCE] weston 1.8.0" (https://lists.freedesktop.org/archives/wayland-devel/2015-June/022416.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20200815084805/https://lists.freedesktop.org/archives/wayland-devel/2015-June/022416.html) from the original on 15 August 2020. Retrieved 14 January 2020.
163. Harrington, Bryce (21 September 2015). "[ANNOUNCE] wayland 1.9.0" (https://lists.freedesktop.org/archives/wayland-devel/2015-September/024302.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20201025222349/https://lists.freedesktop.org/archives/wayland-devel/2015-September/024302.html) from the original on 25 October 2020. Retrieved 14 January 2020.

164. Harrington, Bryce (21 September 2015). "[ANNOUNCE] weston 1.9.0" (https://lists.freedesktop.org/archives/wayland-devel/2015-September/024303.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20200921044115/https://lists.freedesktop.org/archives/wayland-devel/2015-September/024303.html) from the original on 21 September 2020. Retrieved 14 January 2020.
165. Ådahl, Jonas (24 November 2015). "[ANNOUNCE] wayland-protocols 1.0" (https://lists.freedesktop.org/archives/wayland-devel/2015-November/025759.html). *wayland-devel* (Mailing list).
166. Harrington, Bryce (17 February 2016). "[ANNOUNCE] wayland 1.10.0" (https://lists.freedesktop.org/archives/wayland-devel/2016-February/027038.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20160217083326/https://lists.freedesktop.org/archives/wayland-devel/2016-February/027038.html) from the original on 17 February 2016. Retrieved 17 February 2016.
167. Harrington, Bryce (17 February 2016). "[ANNOUNCE] weston 1.10.0" (https://lists.freedesktop.org/archives/wayland-devel/2016-February/027039.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20160224031559/https://lists.freedesktop.org/archives/wayland-devel/2016-February/027039.html) from the original on 24 February 2016. Retrieved 17 February 2016.
168. Nestor, Marius (18 February 2016). "Wayland 1.10 Display Server Officially Released, Wayland 1.11 Arrives in May 2016" (https://news.softpedia.com/news/wayland-1-10-display-server-officially-released-wayland-1-11-arrives-in-may-2016-500558.shtml). *Softpedia*. Archived (https://web.archive.org/web/20200114134844/https://news.softpedia.com/news/wayland-1-10-display-server-officially-released-wayland-1-11-arrives-in-may-2016-500558.shtml) from the original on 14 January 2020. Retrieved 14 January 2020.
169. Larabel, Michael (16 February 2016). "Wayland 1.10 Officially Released" (https://www.phoronix.com/scan.php?page=news_item&px=Wayland-1.10-Released). *Phoronix.com*. Archived (https://web.archive.org/web/20170621204439/http://www.phoronix.com/scan.php?page=news_item&px=Wayland-1.10-Released) from the original on 21 June 2017. Retrieved 17 April 2016.
170. Ådahl, Jonas (16 February 2016). "[ANNOUNCE] wayland-protocols 1.1" (https://lists.freedesktop.org/archives/wayland-devel/2016-February/027029.html). *wayland-devel* (Mailing list).
171. Ådahl, Jonas (23 May 2016). "[ANNOUNCE] wayland-protocols 1.4" (https://lists.freedesktop.org/archives/wayland-devel/2016-May/029014.html). *wayland-devel* (Mailing list).
172. Harrington, Bryce (1 June 2016). "[ANNOUNCE] wayland 1.11.0" (https://lists.freedesktop.org/archives/wayland-devel/2016-June/029163.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20160601085111/https://lists.freedesktop.org/archives/wayland-devel/2016-June/029163.html) from the original on 1 June 2016. Retrieved 1 June 2016.
173. Harrington, Bryce (1 June 2016). "[ANNOUNCE] weston 1.11.0" (https://lists.freedesktop.org/archives/wayland-devel/2016-June/029164.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20161008164235/https://lists.freedesktop.org/archives/wayland-devel/2016-June/029164.html) from the original on 8 October 2016. Retrieved 1 June 2016.
174. Ådahl, Jonas (22 July 2016). "[ANNOUNCE] wayland-protocols 1.5" (https://lists.freedesktop.org/archives/wayland-devel/2016-July/030211.html). *wayland-devel* (Mailing list).
175. Ådahl, Jonas (15 August 2016). "[ANNOUNCE] wayland-protocols 1.7" (https://lists.freedesktop.org/archives/wayland-devel/2016-August/030606.html). *wayland-devel* (Mailing list).
176. Harrington, Bryce (21 September 2016). "[ANNOUNCE] wayland 1.12.0" (https://lists.freedesktop.org/archives/wayland-devel/2016-September/031121.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20170402020659/https://lists.freedesktop.org/archives/wayland-devel/2016-September/031121.html) from the original on 2 April 2017. Retrieved 21 September 2016.

177. Harrington, Bryce (21 September 2016). "[ANNOUNCE] weston 1.12.0" (https://lists.freedesktop.org/archives/wayland-devel/2016-September/031123.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20161008110903/https://lists.freedesktop.org/archives/wayland-devel/2016-September/031123.html) from the original on 8 October 2016. Retrieved 21 September 2016.
178. Harrington, Bryce (21 February 2017). "[ANNOUNCE] wayland 1.13.0" (https://lists.freedesktop.org/archives/wayland-devel/2017-February/033193.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20170224121408/https://lists.freedesktop.org/archives/wayland-devel/2017-February/033193.html) from the original on 24 February 2017. Retrieved 30 April 2017.
179. Harrington, Bryce (25 February 2017). "[ANNOUNCE] weston 2.0.0" (https://lists.freedesktop.org/archives/wayland-devel/2017-February/033231.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20170302071023/https://lists.freedesktop.org/archives/wayland-devel/2017-February/033231.html) from the original on 2 March 2017. Retrieved 30 April 2017.
180. Ådahl, Jonas (31 July 2017). "[ANNOUNCE] wayland-protocols 1.10" (https://lists.freedesktop.org/archives/wayland-devel/2017-July/034706.html). *wayland-devel* (Mailing list).
181. Harrington, Bryce (8 August 2017). "[ANNOUNCE] wayland 1.14.0" (https://lists.freedesktop.org/archives/wayland-devel/2017-August/034748.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20181020173506/https://lists.freedesktop.org/archives/wayland-devel/2017-August/034748.html) from the original on 20 October 2018. Retrieved 25 March 2018.
182. Harrington, Bryce (8 August 2017). "[ANNOUNCE] weston 3.0.0" (https://lists.freedesktop.org/archives/wayland-devel/2017-August/034749.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20181020180931/https://lists.freedesktop.org/archives/wayland-devel/2017-August/034749.html) from the original on 20 October 2018. Retrieved 25 March 2018.
183. Ådahl, Jonas (11 October 2017). "[ANNOUNCE] wayland-protocols 1.11" (https://lists.freedesktop.org/archives/wayland-devel/2017-October/035397.html). *wayland-devel* (Mailing list).
184. Ådahl, Jonas (14 February 2018). "[ANNOUNCE] wayland-protocols 1.13" (https://lists.freedesktop.org/archives/wayland-devel/2018-February/036992.html). *wayland-devel* (Mailing list).
185. Foreman, Derek (9 April 2018). "[ANNOUNCE] wayland 1.15.0" (https://lists.freedesktop.org/archives/wayland-devel/2018-April/037767.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20180410072116/https://lists.freedesktop.org/archives/wayland-devel/2018-April/037767.html) from the original on 10 April 2018. Retrieved 15 April 2018.
186. Foreman, Derek (9 April 2018). "[ANNOUNCE] weston 4.0.0" (https://lists.freedesktop.org/archives/wayland-devel/2018-April/037768.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20180410202212/https://lists.freedesktop.org/archives/wayland-devel/2018-April/037768.html) from the original on 10 April 2018. Retrieved 15 April 2018.
187. Ådahl, Jonas (7 May 2018). "[ANNOUNCE] wayland-protocols 1.14" (https://lists.freedesktop.org/archives/wayland-devel/2018-May/038103.html). *wayland-devel* (Mailing list).
188. Ådahl, Jonas (30 July 2018). "[ANNOUNCE] wayland-protocols 1.16" (https://lists.freedesktop.org/archives/wayland-devel/2018-July/039202.html). *wayland-devel* (Mailing list).
189. Foreman, Derek (24 August 2018). "[ANNOUNCE] wayland 1.16.0" (https://lists.freedesktop.org/archives/wayland-devel/2018-August/039358.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20180828103108/https://lists.freedesktop.org/archives/wayland-devel/2018-August/039358.html) from the original on 28 August 2018. Retrieved 30 November 2018.

190. Foreman, Derek (24 August 2018). "[ANNOUNCE] weston 5.0.0" (https://lists.freedesktop.org/archives/wayland-devel/2018-August/039359.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20180825180241/https://lists.freedesktop.org/archives/wayland-devel/2018-August/039359.html) from the original on 25 August 2018. Retrieved 30 November 2018.
191. Ådahl, Jonas (12 November 2018). "[ANNOUNCE] wayland-protocols 1.17" (https://lists.freedesktop.org/archives/wayland-devel/2018-November/039662.html). *wayland-devel* (Mailing list).
192. Foreman, Derek (28 March 2019). "[ANNOUNCE] wayland 1.17.0" (https://lists.freedesktop.org/archives/wayland-devel/2019-March/040335.html) (Mailing list). Archived (https://web.archive.org/web/20190325203121/https://lists.freedesktop.org/archives/wayland-devel/2019-March/040335.html) from the original on 25 March 2019. Retrieved 17 June 2019.
193. Foreman, Derek (21 March 2019). "[ANNOUNCE] weston 6.0.0" (https://lists.freedesktop.org/archives/wayland-devel/2019-March/040356.html) (Mailing list). Archived (https://web.archive.org/web/20190331194923/https://lists.freedesktop.org/archives/wayland-devel/2019-March/040356.html) from the original on 31 March 2019. Retrieved 17 June 2019.
194. Ådahl, Jonas (25 July 2019). "[ANNOUNCE] wayland-protocols 1.18" (https://lists.freedesktop.org/archives/wayland-devel/2019-July/040756.html). *wayland-devel* (Mailing list).
195. Ser, Simon (11 February 2020). "[ANNOUNCE] wayland 1.18" (https://lists.freedesktop.org/archives/wayland-devel/2020-February/041207.html) (Mailing list). Archived (https://web.archive.org/web/20210401042718/https://lists.freedesktop.org/archives/wayland-devel/2020-February/041207.html) from the original on 1 April 2021. Retrieved 12 February 2020.
196. Ser, Simon (23 August 2019). "[ANNOUNCE] weston 7.0.0" (https://lists.freedesktop.org/archives/wayland-devel/2019-August/040812.html) (Mailing list). Archived (https://web.archive.org/web/20190825033310/https://lists.freedesktop.org/archives/wayland-devel/2019-August/040812.html) from the original on 25 August 2019. Retrieved 22 October 2019.
197. Ser, Simon (24 January 2020). "[ANNOUNCE] weston 8.0.0" (https://lists.freedesktop.org/archives/wayland-devel/2020-January/041147.html) (Mailing list). Retrieved 12 February 2020.
198. Ser, Simon (4 September 2020). "[ANNOUNCE] weston 9.0.0" (https://lists.freedesktop.org/archives/wayland-devel/2020-September/041595.html) (Mailing list). Archived (https://web.archive.org/web/20200929211532/https://lists.freedesktop.org/archives/wayland-devel/2020-September/041595.html) from the original on 29 September 2020. Retrieved 30 October 2020.
199. Ådahl, Jonas (29 February 2020). "[ANNOUNCE] wayland-protocols 1.19" (https://lists.freedesktop.org/archives/wayland-devel/2020-February/041267.html). *wayland-devel* (Mailing list).
200. Ådahl, Jonas (29 February 2020). "[ANNOUNCE] wayland-protocols 1.20" (https://lists.freedesktop.org/archives/wayland-devel/2020-February/041269.html). *wayland-devel* (Mailing list).
201. Ser, Simon (27 January 2021). "[ANNOUNCE] wayland 1.19.0" (https://lists.freedesktop.org/archives/wayland-devel/2021-January/041692.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20210201091228/https://lists.freedesktop.org/archives/wayland-devel/2021-January/041692.html) from the original on 1 February 2021. Retrieved 27 January 2021.
202. Ådahl, Jonas (30 April 2021). "[ANNOUNCE] wayland-protocols 1.21" (https://lists.freedesktop.org/archives/wayland-devel/2021-April/041815.html). *wayland-devel* (Mailing list).
203. Ådahl, Jonas (23 November 2021). "[ANNOUNCE] wayland-protocols 1.24" (https://lists.freedesktop.org/archives/wayland-devel/2021-November/042039.html). *wayland-devel* (Mailing list).

204. Ser, Simon (27 January 2021). "[ANNOUNCE] wayland 1.20.0" (https://lists.freedesktop.org/archives/wayland-devel/2021-December/042064.html). *wayland-devel* (Mailing list). Archived (https://web.archive.org/web/20211211142112/https://lists.freedesktop.org/archives/wayland-devel/2021-December/042064.html) from the original on 11 December 2021. Retrieved 27 January 2021.
205. Ser, Simon (1 February 2022). "[ANNOUNCE] weston 10.0.0" (https://lists.freedesktop.org/archives/wayland-devel/2022-February/042103.html) (Mailing list). Archived (https://web.archive.org/web/20220204232432/https://lists.freedesktop.org/archives/wayland-devel/2022-February/042103.html) from the original on 4 February 2022. Retrieved 12 February 2022.
206. Vlad, Marius (2 August 2023). "[ANNOUNCE] weston 10.0.5" (https://lists.freedesktop.org/archives/wayland-devel/2023-August/042925.html). *wayland-devel* (Mailing list).
207. Ådahl, Jonas (28 January 2022). "[ANNOUNCE] wayland-protocols 1.25" (https://lists.freedesktop.org/archives/wayland-devel/2022-January/042102.html). *wayland-devel* (Mailing list).
208. Ser, Simon (30 June 2022). "[ANNOUNCE] wayland 1.21.0" (https://lists.freedesktop.org/archives/wayland-devel/2022-June/042268.html). *wayland-devel* (Mailing list).
209. Ser, Simon (22 September 2022). "[ANNOUNCE] weston 11.0.0" (https://lists.freedesktop.org/archives/wayland-devel/2022-September/042410.html). *wayland-devel* (Mailing list).
210. Vlad, Marius (2 August 2023). "[ANNOUNCE] weston 11.0.3" (https://lists.freedesktop.org/archives/wayland-devel/2023-August/042924.html). *wayland-devel* (Mailing list).
211. Ådahl, Jonas (7 July 2022). "[ANNOUNCE] wayland-protocols 1.26" (https://lists.freedesktop.org/archives/wayland-devel/2022-July/042270.html). *wayland-devel* (Mailing list).
212. Ådahl, Jonas (29 November 2022). "[ANNOUNCE] wayland-protocols 1.31" (https://lists.freedesktop.org/archives/wayland-devel/2022-November/042524.html). *wayland-devel* (Mailing list).
213. Ser, Simon (4 April 2023). "[ANNOUNCE] wayland 1.22.0" (https://lists.freedesktop.org/archives/wayland-devel/2023-April/042647.html). *wayland-devel* (Mailing list). Retrieved 9 April 2023.
214. Vlad, Marius (17 May 2023). "[ANNOUNCE] weston 12.0.0" (https://lists.freedesktop.org/archives/wayland-devel/2023-May/042720.html). *wayland-devel* (Mailing list).
215. "Weston 12.0.5 release" (https://lists.freedesktop.org/archives/wayland-devel/2025-April/044163.html). *Wayland-devel mailing list*. freedesktop.org. April 2025. Retrieved 14 August 2025.
216. Vlad, Marius (28 November 2023). "[ANNOUNCE] weston 13.0.0" (https://lists.freedesktop.org/archives/wayland-devel/2023-November/043326.html). *wayland-devel* (Mailing list).
217. "Weston 13.0.4 release" (https://lists.freedesktop.org/archives/wayland-devel/2025-April/044159.html). *Wayland-devel mailing list*. freedesktop.org. April 2025. Retrieved 14 August 2025.
218. Ådahl, Jonas (3 July 2023). "[ANNOUNCE] wayland-protocols 1.32" (https://lists.freedesktop.org/archives/wayland-devel/2023-July/042836.html). *wayland-devel* (Mailing list).
219. Ådahl, Jonas (26 April 2024). "[ANNOUNCE] wayland-protocols 1.36" (https://lists.freedesktop.org/archives/wayland-devel/2024-April/043581.html). *wayland-devel* (Mailing list).
220. Ser, Simon (30 May 2024). "[ANNOUNCE] wayland 1.23.0" (https://lists.freedesktop.org/archives/wayland-devel/2024-May/043636.html). *wayland-devel* (Mailing list).
221. Vlad, Marius (4 September 2024). "[ANNOUNCE] weston 14.0.0" (https://lists.freedesktop.org/archives/wayland-devel/2024-September/043776.html). *wayland-devel* (Mailing list).
222. Ådahl, Jonas (31 August 2024). "[ANNOUNCE] wayland-protocols 1.37" (https://lists.freedesktop.org/archives/wayland-devel/2024-August/043774.html). *wayland-devel* (Mailing list).
223. "[ANNOUNCE] wayland-protocols 1.45" (https://lore.freedesktop.org/wayland-devel/aEw0AP7h6T8l11ug@gmail.com/T/). *Wayland-devel mailing list (lore.freedesktop.org)*. freedesktop.org. 13 June 2025. Retrieved 14 August 2025.

External links

- Official website (https://wayland.freedesktop.org/)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Wayland_(protocol)&oldid=1307351891"