

# Pure Wayland toolkit prototype

XWayland server provides limited capabilities for X11 desktop applications (see [X11 Application Support](#) and [JDK-8269245](#)). In order to get full support of the desktop features, we need to implement a pure Wayland client toolkit for java. Wayland architecture in [many ways](#) differs from X11, so we cannot reuse XAWT even for basic capabilities. The new toolkit should be implemented from scratch. Here are some major chunks of work:

- Event handling
  - Dispatch native events on EDT, to avoid potential race conditions when the state is updated both from EDT and toolkit thread. See the full proposal [here](#)
- Graphics devices support
  - Onscreen/offscreen Wayland surface management
  - Adopt OGL pipeline for rendering on Wayland surfaces
  - Implement a new rendering pipeline based on Vulkan (for better performance)
- `java.awt.Robot`
  - Sending input events
  - Reading screen data (at least current java application windows)
- Client-side decorations for windows
  - Swing internal frames rendering code can be reused
- Hardware acceleration using Vulkan.

The prototype implementation of the pure Wayland toolkit ([JDK-8281970](#)) based on OpenJDK 21 source base can be found in the Wakefield repository ([jdk 21.0.1-wayland](#) branch). It has two rendering pipelines:

- Pure software loops (enabled by default)
- Hardware accelerated rendering using Vulkan (enabled by providing `-Dsun.java2d.vulkan=True|true`)

```
java -Dawt.toolkit.name=WLTToolkit -Dsun.java2d.vulkan=True J2Ddemo.jar
```

[blocked URL](#)

Here is an updated build instruction for building `jdk21.0.1-wayland` branch (on ubuntu 24.04):

```
$ sudo wget -qO- https://packages.lunarg.com/lunarg-signing-key-pub.asc | tee /etc/apt/trusted.gpg.d/lunarg.asc
$ sudo wget -qO /etc/apt/sources.list.d/lunarg-vulkan-noble.list https://packages.lunarg.com/vulkan/lunarg-vulkan-noble.list
$ sudo apt update
$ sudo apt-get -y install openjdk-21-jdk file zip unzip autoconf make build-essential libx11-dev libxext-dev libxrender-dev \
libxtst-dev libxt-dev libxrandr-dev libcups2-dev libfontconfig1-dev libasound2-dev libwayland-dev \
libxkbcommon-x11-0 vulkan-sdk vulkan-utility-libraries-dev
$ sh configure --with-vulkan
$ make images
```