

# Open Unsolved Problems

## java.awt.Robot taking screenshots

*Proposed solution is [org.freedesktop.portal.Screenshot](#). It is still a DBUS API, but at least it seems to be a public one.*

It is doable as of now, but not in a standard and convenient way.

It can be done using some shell specific APIs, e.g. [org.gnome.Shell.Screenshot DBUS API](#) (which will not work for KDE, it has it's has another API)

It has several drawbacks:

- There is no direct access to the screenshot data. Screenshot is saved to `png` file, after that you have to read it, decode it.
- Filesystem footprint. You need to save it to some file(its filesystem can be slow). It can be workarounded by using shared memory object though.
- Using DBUS API in this case is really slow. For instance, just a sync DBUS call to take a 1000x1000 screenshot and save it to the shared memory object file is ~14 time slower(it is just saving, without decoding and extracting data) comparing to full X11 implementation.

So, it would be great to get some standard API to get screenshot data in memory. I see that there is an [open bug](#) against this issue.

## java.awt.Robot emulating keyboard/mouse events

*Looks like the solution for native Wayland client will be [libEI](#). Its shipping time(even estimated) is not yet known.*

It mostly works for X11 compatibility mode(except when trying to reach outside XWayland and windows are not restacked on emulated mouse click).

Looks like the solution for native Wayland client will be [libEI](#)

## X11 compatibility: HiDPI.

*No solution yet.*

Quote from Maxim Kartashev's e-mail:

```
There's a quality-of-service problem with running via the compatibility layer as under certain circumstances native X windows look blurry.
```

```
Users with small(ish) HiDPI displays tend to enable fractional scaling and with that enabled (regardless of the actual scale), XWayland pretends that the screen size is smaller and then pixel-stretches the resulting window according to the global scale. This works as a temporary solution, but people get quickly tired of looking at text that is blurry.
```

```
See https://gitlab.gnome.org/GNOME/mutter/-/issues/402 and https://github.com/swaywm/sway/issues/5917 for some more details.
```

## X11 compatibility: Popup menu dismiss in X11 compatibility mode.

*Can be not perfectly workarounded by dismissing menu on focus lost event*

We are using `XGrabPointer` to get mouse input and dismiss popup menus on mouse click.

Obviously, it does not work outside of XWayland server.

E.g. if we have shown some popup menu, it will be closed upon clicking on any of XWayland's windows.

But it will not be closed if we click on some other window from Wayland.

```

diff --git a/src/java.desktop/unix/classes/sun/awt/X11/XPopupMenuPeer.java b/src/java.desktop/unix/classes
/sun/awt/X11/XPopupMenuPeer.java
index a19f56249ae..db88ef49f37 100644
--- a/src/java.desktop/unix/classes/sun/awt/X11/XPopupMenuPeer.java
+++ b/src/java.desktop/unix/classes/sun/awt/X11/XPopupMenuPeer.java
@@ -111,6 +111,16 @@ public class XPopupMenuPeer extends XMenuWindow implements PopupMenuPeer {
    // Get menus from the target.
    Vector<MenuItem> targetItemVector = getMenuTargetItems();
    if (targetItemVector != null) {
+
+ //TODO: add focus listener only for XWayland
+ target.addFocusListener(new FocusAdapter() {
+
+     @Override
+     public void focusLost(FocusEvent e) {
+         target.removeFocusListener(this);
+         if (isShowing()) {
+             hide();
+         }
+     }
+ });
    reloadItems(targetItemVector);
    //Fix for 6287092: JCK15a: api/java_awt/interactive/event/EventTests.html#EventTest0015 fails,
mustang

```

For a first look this workaround seems to work reliably except only one case:

clicking on window's title containing origin component does not lead to focus change, thus we don't hiding a popup.

I see a behavior similar to this workaround in some other third party application running on XWayland.

## X11 compatibility: fake configure event for XRandR emulation

*Should be resolved with this [MR](#).*

We are not getting updates from the system after "changing" screen resolution via `XRRSetScreenConfigAndRate`.

Excerpt from Olivier's mail:

Also worth noting that the XRandR emulation is per window/X11 client, whereas the root window is shared between all X11 clients, but maybe we could send a fake `ConfigureNotify` event to the given client, I would need to check if that's doable.

This notification would be helpful.

## X11 compatibility: Wayland crash on huge window

*Resolved by [1](#) and [2](#). Waiting for its propagation to Linux distros.*

```

// gcc hugeFrame.c -o hugeFrame -lX11 && ./hugeFrame
#include <X11/Xlib.h>
#include <unistd.h>

static Display* display;

int main(void) {
    int width = 22000;
    int height = 25000;

    display = XOpenDisplay(NULL);

    Window win = XCreateSimpleWindow(display, DefaultRootWindow(display), 0, 0, width, height, 0, 0L, 0L);
    XMapWindow(display, win);
    XFlush(display);

    sleep(1);
}

```

`/var/log/syslog` shows:

```
gnome-shell[1248]: WL: error in client communication (pid 1248)
gnome-shell[1298]: (EE)
gnome-shell[1298]: Fatal server error:
gnome-shell[1298]: (EE) wl_shm@5: error 1: invalid size (-2094967296)
gnome-shell[1298]: (EE)
```

where -2094967296 looks like an integer overflow of  $4 * 22000 * 25000$

So it fails to create a [pool](#), but doesn't handle it gracefully.

## X11 compatibility: crash when mapping a lot of windows

*No solution yet*

GUI are crashing if you are trying to map a lot (> ~260) of windows at once.

Filed as [xorg/xserver/-/issues/1222](https://bugzilla.gnome.org/show_bug.cgi?id=1222).

```
// gcc windowSpawnDoS.c -o windowSpawnDoS -lX11 && ./windowSpawnDoS
#include <X11/Xlib.h>
#include <unistd.h>

int main(void) {
    XInitThreads();
    Display* dpy = XOpenDisplay(NULL);

    Window root = DefaultRootWindow(dpy);
    for (int i = 0; i < 500; ++i) {
        Window ww = XCreateSimpleWindow(dpy, root, 50, 50, 50, 50, 0, 0, 0);
        XMapWindow(dpy, ww);
    }

    for(;;) {
        XEvent e;
        XNextEvent(dpy, &e);
    }
}
```

It may be faced when running some tests which are trying to map a lot of windows, e.g. one for every GraphicsConfiguration.

If you have 210 GraphicsConfigurations per display on Xwayland, then running such test in 2 display configuration will lead to crash.