Automated testing for mobile images using **GNOME**

Sam Thursfield

Tech Stacks for Open Source Smartphones

Android

Examples:

- CalyxOS
- /e/os
- GrapheneOS
- LineageOS

GTK + GNOME

Examples:`

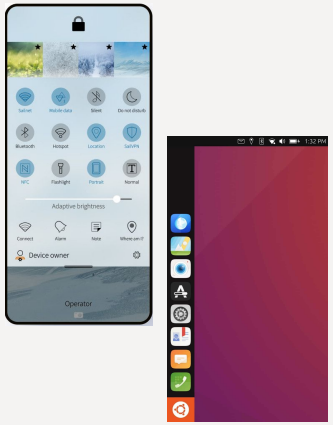
- postmarketOS + Phosh
- postmarketOS + upstream GNOME

Qt + KDE

Examples:`

- postmarketOS + Plasma Mobile

Qt + custom widgets



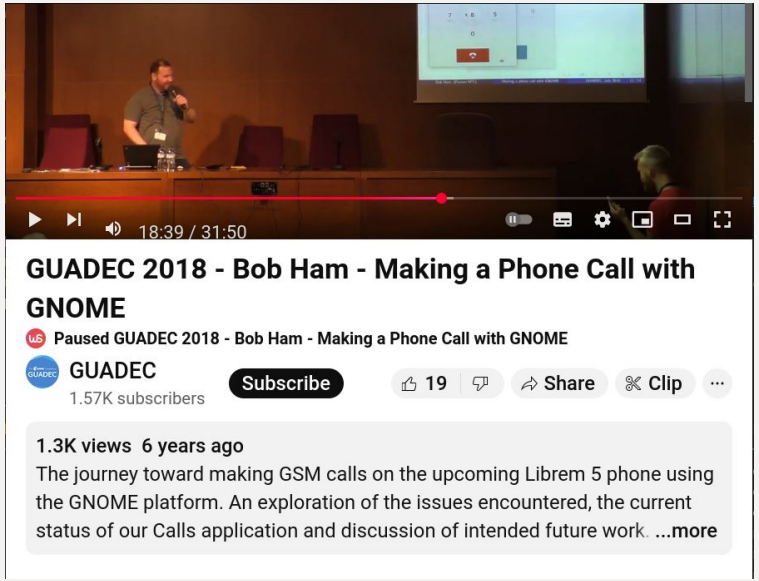
Examples:`

- Sailfish OS
- Lomiri (Ubuntu Touch)

Why use **GTK + GNOME** on your phone?

1. The same bugs you have on the desktop – **now in your pocket!**
2. No need to write code in Java or C++.
3. You can use it **make phone calls**, in principle.
4. Volunteer-driven development. You just volunteered to join the QA team!

GUADEC 2018 – Bob Ham



What is mobile **GNOME?**

Platform libraries and services:

libadwaita developers created beautiful **adaptive UI components** using GTK.

Core apps:

GNOME core app designers and developers built modern **adaptive UIs**.

GNOME Shell:

Work in progress – see gnome-shell-mobile.

Phosh also integrates with GNOME.

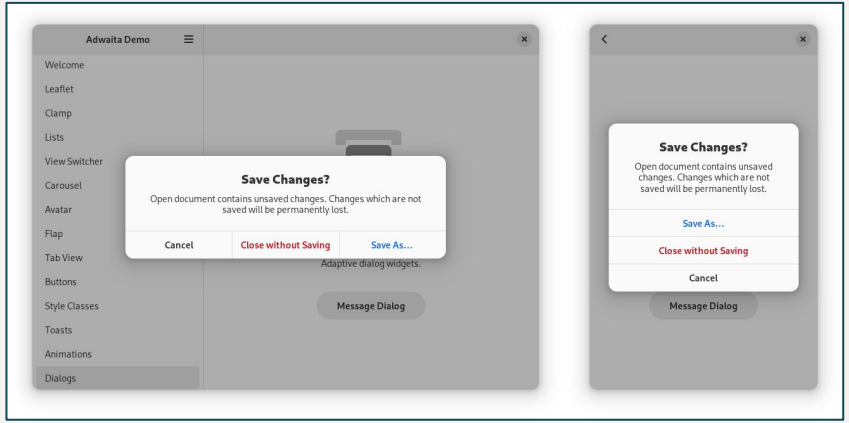


Image: Alice Mikhaylenko

Small Size Handling

The smallest recommended display size for GNOME on desktop is currently 1024×600px, and this size should be supported by all apps. Apps that are appropriate for a phone form factor should scale down to 360×294px.

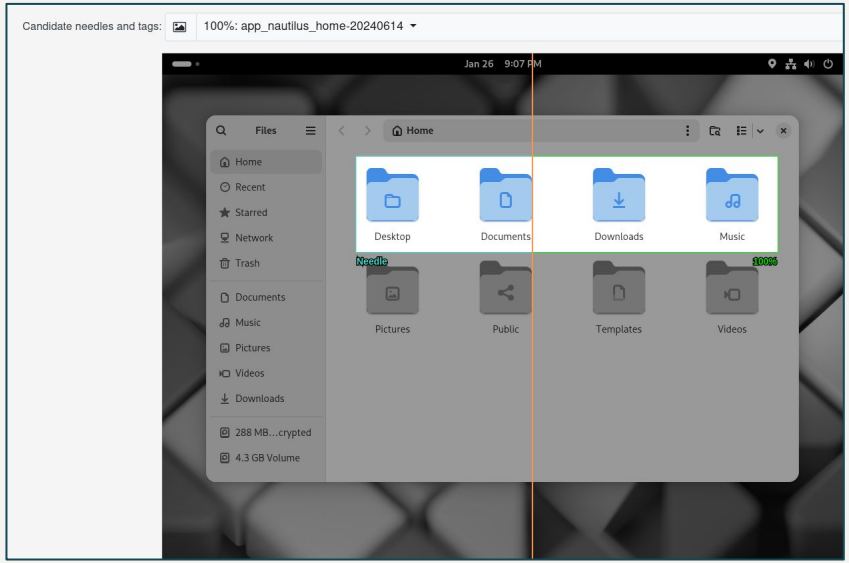
From GNOME Human Interface Guidelines

Automated end-to-end testing of **GNOME**

In 2021, an internal Codethink project set up **openQA testing** for GNOME OS.

On every change to gnome-build-meta, Gitlab CI builds and launches a VM.

The openqa-tests navigate the UI like an end user, using fuzzy image matching and a virtual keyboard and mouse.



Automated end-to-end testing of **GNOME** mobile

In 2023-2024, I ran an Outreachy internship around the openQA tests.

Adding a gnome_mobile testsuite.
How hard can it be?

- ✗ Setting screen size of the VM to 360 × 720: Doesn't work due to an **openQA / os-autoinst limitation**.
- ✗ Setting screen size of the VM to 720 × 1440, with 2x scaling: Doesn't work due to a **Mutter limitation**.
- ✗ Setting pixel density and chassis type in QEMU: Possible, but difficult.

(We did add a mobile test suite in the end... and several other test suites).

Pixel density

In 2024, Adrien Plazas at Codethink investigated again.

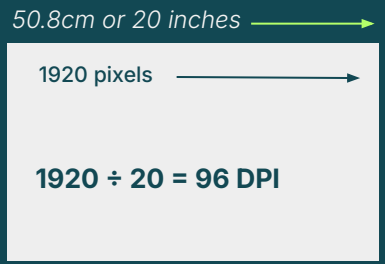
QEMU's *virtio-vga* driver can send **display identification data** (EDID) including the physical width & height in centimeters.

There's no way to **set** physical dimensions of the virtual screen. QEMU's default is based around a 100 DPI desktop monitor.

Mutter uses the EDID to decide an initial scale factor (see `calculate_scale()` in meta-monitor.c).

*We could override the EDID **inside** the system under test as a workaround – but this is not a long term solution!*

Pixel density of a 24 inch monitor



Pixel density of hacker-friendly phones

Device	Diagonal	Resolution	Density	UI Scale
Librem 5	5.7″	720 × 1440px	282 ppi	200%
PinePhone	5.95″	720 × 1440px	270 ppi	200%
PinePhone Pro	6″	720 × 1440px	268 ppi	200%
OnePlus 6	6.28″	1080 × 2280px	401 ppi	300%
OnePlus 6T	6.41″	1080 × 2340px	402 ppi	300%

SMBIOS and chassis type

SMBIOS data structures provide **hardware info** to the OS.

It will be fun to read the SMBIOS standard! Let's check data structure type 3: **System Enclosure or Chassis.**

QEMU's -smbios argument lets us set other fields but not the chassis type:

```
-smbios type=3[,manufacturer=str][,version=str][,serial=str][,asset=str][,sku=str]
```

Specify SMBIOS type 3 fields

gnome-settings-daemon reads this setting and tweaks OS behaviour.

1001 **7.4.1 System Enclosure or Chassis Types**

1002 Table 17 shows the byte values for the System Enclosure or Chassis Types field.

1003 NOTE Refer to 6.3 for the CIM properties associated with this enumerated value.

1004 **Table 17 – System Enclosure or Chassis Types**

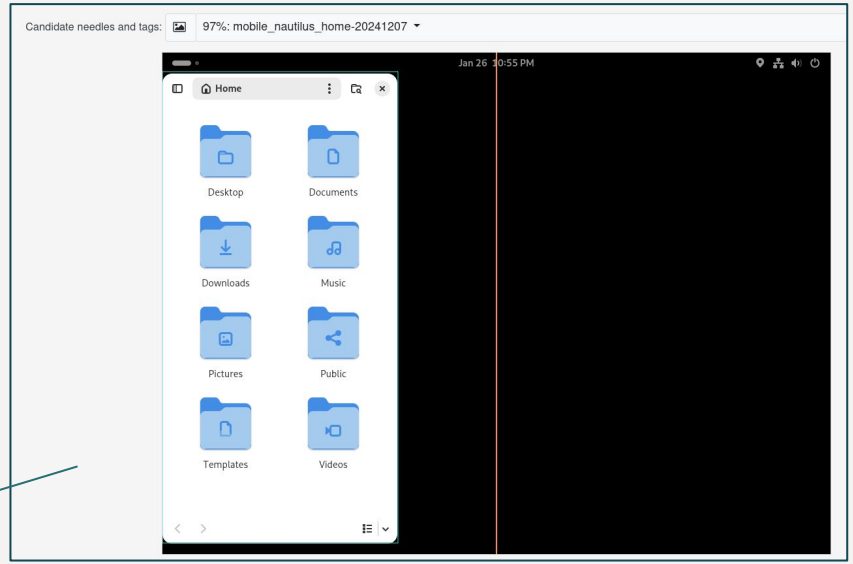
Byte Value	Meaning
01h	Other
02h	Unknown
03h	Desktop
04h	Low Profile Desktop
05h	Pizza Box
06h	Mini Tower
07h	Tower
08h	Portable
09h	Laptop
0Ah	Notebook
0Bh	Hand Held
0Ch	Docking Station
0Dh	All in One
0Eh	Sub Notebook
0Fh	Space-saving
10h	Lunch Box
11h	Main Server Chassis
12h	Expansion Chassis
13h	SubChassis
14h	Bus Expansion Chassis
15h	Peripheral Chassis

Future work

1. QEMU patches.
2. Multitouch gesture input. Requires support in openQA to make this “easy to use”.
3. Hardware testing. Requires volunteers to host and maintain hardware test rigs.
4. Tests maintained as part of the application itself.
5. Come to my talk “How to push your testing upstream” on Sunday 2nd Feb, Distributions devroom, 14:00.

The **gnome_mobile** testsuite

gnome_disable_update_notification 1s	passed	
gnome_desktop 4s	passed	
mobile_change_background_colour 3s	passed	
app_gnome_calculator 30s	passed	
app_gnome_characters 19s	passed	
app_gnome_clocks 21s	passed	
app_gnome_console 29s	passed	
app_gnome_contacts 17s	passed	
app_gnome_font_viewer 18s	passed	
app_gnome_text_editor 19s	passed	
app_loupe 20s	passed	
app_nautilus 20s	passed	
app_simple_scan 19s	passed	
app_snapshot 18s	passed	



Today's workaround: *Start in desktop form factor, and manually resize the app windows to 360×294.*

Get involved

Linux QA monthly open call:

- **Second Thursday** of every month
- https://gitlab.gnome.org/GNOME/openqa-tests/-/wikis/QA-testing-monthly-call

Matrix: #mobile:gnome.org

FOSS on mobile devices devroom - **Saturday afternoon, H building.**

Thanks

Dorothy Kabarozi **Tanju Acheleke** **Adrien Plazas**

... and everyone out there contributing to open source!

Thank You.

Codethink Ltd.

3rd Floor Dale House,
35 Dale Street,
MANCHESTER,
M1 2HF,
United Kingdom