Wayland for Qt Application Developers
About Me

David Edmundson
This Talk
As an app developer, why should you care?
Why should you care - the positive

- More detailed input events
- Faster rendering
- Reduced power consumption
- General Amazingness...
Why should you care - the negative
Why should you care - the negative

You won't have a choice
if (app.platformName() == "wayland") {
    QMessageBox::critical(0, i18nc("title:window", "Fatal Error"), i18n("App does not support the Wayland platform. Use XWayland to run App on Wayland. App will close now.");
    return -1;
}
Current State
Qt abstracts, so in theory things should just work
The difference between theory and practice is that in theory they're both the same.
Bugs - who's at fault

- Deliberate Wayland behavioural differences
- Bugs in your app
- QtWayland
- The compositor
- Missing specifications
Deliberate wayland behavioural differences
Making X calls

- Any unguarded use of libX11/libXCB obviously fails
- Use appropriate abstraction layers where possible
Eavesdropping events

- Wayland clients only get input when they have focus
- Custom idle detection or global shortcut detection by hand won't work
- KIdleTime and KGlobalAccel work out of the box
Grabbing other window content

- QScreen::grabWindow won't work
- Replace with DBus requests to xdg-desktop-portal
Global positioning

- Wayland windows don't know where they are, nor can they set where they are.
- One of the biggest porting challenges, but there are options:
Xdg Positioner

- Semantically state where a popup should be relative to the parent window
- Can even provide hints for how to handle constraints

Pros:
- Lots of potential

Cons:
- No Qt API available to access all the features
Plasmashell Protocols

• Set exact position + layer in global coordinates

• Example:
  • Plasma panels

• Cons:
  • Not universally supported
  • Access is restricted
  • Implementation is poor
  • We are trying to phase it out
Layershell

- Semantic position relative to a screen

- Example:
  - KSplash

- Pros:
  - All features fully exposed in a KDE library
  - Somewhat standardised

- Cons:
  - You can't have both layer shell and regular windows in the same app.
Popups

- Set a transient parent
- Create in the correct order
- Destroy in the correct order
- Anything else is ILLEGAL!
- QtWayland tries to hide problems, but it's a guessing game
Window activation

• Passing focus from window to another, i.e:
  • Clicking a link and opening a browser
  • Opening a new file from dolphin
  • Restoring from system tray
Window activation - a reminder of X11

- What you should be doing:
  - Create a magic token
  - Pass this token using any other method (dbus, env vars)
  - Import the token into the new client
  - Use QX11Extras/KStartupInfo to set the imported token
  - Only then call QWindow::requestActivate
Window activation - a reminder of X11

- What you should be doing:
  - Create a magic token
  - Pass this token using any other method (dbus, env vars)
  - Import the token into the new client
  - Use QX11Extras/KStartupInfo to set the imported token
  - Only then call QWindow::requestActivate

- Or:
  - Ignore all this and call KwindowSystem::forceActiveWindow #yolo
Window activation - wayland

• What you should be doing:
  • Create a magic
  • Pass a magic string using any other method (dbus, env vars)
  • Import the token into the new client
Window activation - wayland

- Concept is the same, API is subtly different (and easier), see:
  - KWindowSystem::requestXdgActivationToken
  - KWindowSystem::setCurrentXdgActivationToken

- For launching applications
  KIO::*LauncherJob handles export for us
Minimising to the system tray

- QSystemTray or KStatusNotifierItem will continue to work
- You don't have API to know when a window is minimised
Bugs in your app
Bugs in your app

Wayland is not bug-compatible
Bugs - wayland widget window woes

- Every QWidget can be a QWindow
- This makes sense when done deliberately for performance reasons
- X11 window embedding is analogous to Wayland subsurfaces
- This works great in theory when used properly
Bugs - wayland widget window woes

- It's very easy to accidentally turn a QWidget into a window
- Doing so accidentally causes no issues on QtXCB
- It has very odd effects in QtWayland
• Avoid QGLWidget. Use QOpenGLWidget instead

• If you are deliberately using subsurfaces, be sure to enable
  Qt::AA_DontCreateNativeWidgetSiblings
Bugs in your app - Takeaway

Just because an app works fine in X11, doesn't mean there aren't hidden bugs
QtWayland
QtWayland

QtWayland has an impossible job
QtWayland

- Wayland lets you cherry-pick what to support

- QtWayland has two users:
  - Desktop
  - Embedded

- These have divergent code paths
QtWayland

KDE's Contributions
QtWayland - KDE's Contributions

- Well over 50 commits
- 2 Developers gained approver status
- Desktop side is our responsibility
QtWayland - KDE's Contributions

- Well over 50 commits
- 2 Developers gained approver status
- Desktop side is our responsibility
- The Qt maintainers have been amazing
QtWayland - next steps

- We need to have export more native API for wayland specifics
  - Native LayerShell
  - Native window activation
  - Anything that can't be abstracted
QtWayland

BOF this week!
Extending QtWayland
Extending Qt - Additional protocols

- Slightly fragmented landscape in KDE
  - KWayland provides low level API for several additional classes
  - Qt also now provides a mechanism `QtWaylandClientExtension`

- We are in a transitional phase
xtending Qt - QtWaylandClientExtension

- `qtwaylandscanner` is just like `qdbusxml2cpp`
- creates virtuals for events (messages to our client)
- has methods for making requests (messages to the compositor)
- QWaylandClientExtensionTemplate performs the "binding"

- If in doubt, copy the code in kguiaddons/src/recorder and adjust
Missing functionality
Missing functionality

The wayland specs are still evolving
Missing functionality

All specs are being decided by compositor developers. That comes with an inherent bias
Missing functionality - solution

• Get involved!
• Make all problems known
"The best time to plant a tree was 20 years ago.
The second best time is now"
Conclusions
Test, Test, Test, Test, Test, Test, Test
Getting help

- We're here to help
- Please do ping with specific issues
- Use the "wayland" keyword on bugzilla
- Elevate issues to Qt if needed
The end.