Snapping Qt apps

Jesús Soto

KDE Akademy 2023, Thessaloniki GR

My experience with Qt

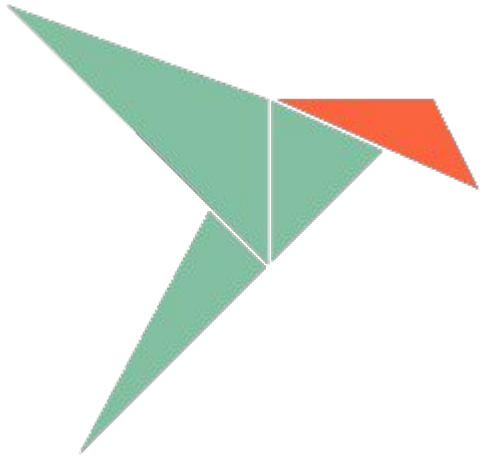
KDE Akademy 2023, Thessaloniki GR

Canonical

Snapping strategies

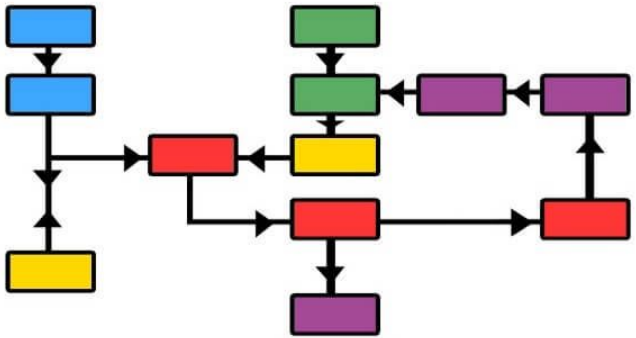
- From source
- From a binary
- From a .deb

In the end, it is just a container running Ubuntu LTS



Stage your dependencies!

- Keep an eye on your base (core##)
- Regular runtime dependencies
- debian/control
- Lint warnings
- ldd your executable
- libQt5Xyz.so.5 => libqt5xyz5 (packages.ubuntu.com)



Remember your environment!

- QT_PLUGIN_PATH
- QML2_IMPORT_PATH
- LD_LIBRARY_PATH (pulseaudio)
- QTWEBENGINEPROCESS_PATH

Add an environment section with your variable definitions

Hardcoded paths & config files

- Try to use environment variables if possible (QTWEBENGINEPROCESS_PATH)
- Layout bindings (alsa, dri drivers, qt itself)

```
layout:
  /usr/lib/$CRAFT_ARCH_TRIPLET/dri:
     bind:
$SNAP/usr/lib/$CRAFT_ARCH_TRIPLET/dri
  /usr/share/alsa:
     bind: $SNAP/usr/share/alsa
  /usr/share/qt5:
     bind: $SNAP/usr/share/qt5
```

Interfaces, plugs and permissions

Interfaces and plugs allow your snap to access resources outside of the sandbox.

Some examples include (not limited to):

- Camera
- Audio record/playback
- Graphics acceleration
- Network
- https://snapcraft.io/docs/supported-interfaces

Some of them require to be connected manually (require approval for auto-connect)

kde-neon extension

The kde-neon extension

This extension helps you snap desktop applications that use Qt5 and/or KDE Frameworks.

https://snapcraft.io/docs/kde-neon-extension

```
apps:
  kcalc:
    extensions:
      - kde-neon
    command: kcalc
    ...
```

Thank you. Questions?