What’s cooking for KDE Frameworks 6
Qt 6 and beyond

2021/06/19, Akademy

Nicolas Fella

nicolas.fella@kde.org
@nicofee
There’s a new Qt in town

- Qt 6 was released end of 2020
- Next major version of KDE Frameworks on the horizon
- New challenges and opportunities

This talk:
- What has happened so far
- What’s going to come
- And why?
Easy porting

• Qt promises easy 5 → 6 transition
• We want the same for KF5 → KF6
• General principles for breaking API changes:
  – Introduce new API now
  – Port users to new API
  – Once all (significant) users are ported deprecate old API
  – Remove old API when branching
• Keep amount of transition-time breakage minimal
Better APIs

- Use Qt or std:: API where appropriate
- Avoid duplication
- Use new C++17 vocabulary
- Address // TODO KF6
- Remove some rarely used stuff
Better Widgets/QtQuick separation

- Many things are entangled with Qt Widgets without needing to
- Improve QML compatibility
- e.g. QWidget → QWindow
- Widgets popups
- Avoiding QtWidgets saves binary size
Reduce dependencies

- Make things easier to deploy
- Save installation size
- Make Frameworks more attractive for third-party users
Better Interface/Implementation separation

- Some Frameworks mix interface and implementation
- Password storage via KWallet
- Sharing via Purpose
- File indexing via Baloo
Better cross-platform support

- Ties into platform/interface topic
- Create such separation where appropriate
- Fill gaps where separation exists
- Avoid DBus, X11, Wayland dependencies
- Add new API to solve recurring problems
Retire some frameworks

- KDELibs4Support
- KHTML
- KDEWebKit
- KJS(Embed)
- Kross
- KMediaPlayer
- KXmlRpcClient
- KInit
A nice Workboard

- Phabricator workboard: https://phabricator.kde.org/project/view/310/
- Currently 384 tasks
- Help needed!
Highlights

- Retire KServiceTypeTrader
- Retire KIconLoader
- Replace KWallet API with QtKeychain
- Rework KRUn
- Finally remove KToolInvocation
- Break up KDEclarative
Questions?