KRunner: Past, Present, and Future
Porting, New Features, and Plugin Distribution

By Alexander Lohnau
Akademy 2023
How does KRunner affect you?

• Most people know it as the standalone executable
  • Launched by Alt+Space/ Alt+F2
  • Invoked when typing on desktop
• KRunner is also a flexible framework
• Powers normal application launchers
• Essential part of desktop experience
Structure

• Who am I and how did I end up in KDE?
• Where we came from - looking back at the KDE4 times
• Major improvements and features
• It all about the plugins - KRunner's ecosystem
• Upcoming changes & improvements in KF6
• Future plans
Who am I and how did I end up in KDE?

• I first tried KDE in 2019, immediately loved it
• Initially played a lot around with applets
• Only later discovered KRunner and all its features
• Played around with third party plugins
• Eventually wrote my first plugin using krunner-bridge
Who am I and how did I end up in KDE?

- Hit some architectural constraints and performance issues
- Rewrote my plugins in C++
- (which I barely knew at the time)
- Most of them are still around on GitHub
My GitHub projects

Pinned

- **krunner-firefox**  Public
  Launch your Firefox profiles from Krunner and register them in the Firefox .desktop file
  - C++  
  - 16  
  - 3

- **EmojiRunner**  Public
  Search for emojis in Krunner and copy/paste them
  - C++  
  - 36  
  - 4

- **KDevelopTemplates**  Public
  This repository contains all my KDevelop templates
  - Shell  
  - 16  
  - 6

- **JetBrainsRunner**  Public
  A Krunner Plugin which allows you to open your recent projects
  - C++  
  - 42  
  - 3

- **JetBrainsDolphinPlugin**  Public
  Dolphin plugin for the JetBrains IDEs
  - C++  
  - 13  
  - 2

- **krunner-kwallet**  Public
  This is a Krunner plugin which allows you to copy/show/edit/add entries from KWallet
  - C++  
  - 4  
  - 2
My origin story - how I ended up in KDE

• Some of my runners required configuration
• Should ideally be done right after installation
• Simple idea: Launch runner's config when opening KCM with argument
  • https://phabricator.kde.org/D26738
• It proved to be quite addictive :}
Where we came from - looking back at the KDE4 times

Source: https://commons.wikimedia.org/wiki/File:Krunner_kde4.3.png#file
Where we came from - looking back at the KDE4 times

• QWidgets based UI
  • Runoptions with match-specific widget
• Help-Button
• Scrollable view
• Scripting support using Kross
  • Python, Ruby, JavaScript
Important features – Help plugin

• Help button KRunner 4 was coded as part of UI
• RunnerSyntax class still existed in KF5 and runners used it
• More flexibility by implementing it as a plugin

• Magic trick: Cast parent to RunnerManager and access other runners
Important features – Help plugin

- spell <search term>
- places
- kill <search term>
- kate <search term>
- g:<search term>
- firefox <search term>

Date and Time
- date
  - Displays the current date
- time
  - Displays the current time
- time <timezone>
  - Displays the current time in a given timezone
- date <timezone>
  - Displays the current date in a given timezone
Important features

- Help plugin
  
  • Available in Kickoff due to being a plugin
Important features – Help plugin

Akademy 2021 Day 3: Took too much time, but at last ... behold KRunner!
Important features – Multiline text

- Used in dictionary and help runner
- Allows usage of Qt's styled text markup
- Used for highlighting and organizing information
- Activated using QueryMatch::setMultiline
Important features – Multiline text

- **suspend**: Lists system suspend (e.g. sleep, hibernate) options and allows them to be activated.

- **screen brightness <percent value>**
  - **dim screen**: Lists screen brightness options or sets it to the brightness defined by the search term; e.g. screen brightness 50 would dim the screen to 50% maximum brightness.
Important features – Testing

- KRunner got more test coverage – framework and individual plugins
- Writing tests that use the runner directly was hard
  - Either manually load a plugin
  - Or build a static lib
- "krunner_configure_test" CMake function to the rescue
Important features – Testing

CMake function for DBus runners and C++ plugins

- `krunner_configure_test(dbusrunnertest testremoterunner DESKTOP_FILE ".../dbusrunnertest.desktop")`
- `krunner_configure_test(datetimerunnertest org.kde.datetime)`

Accompanied by AbstractRunnerTest header:

- `void initProperties()`
- `QProcess *startDBusRunnerProcess(...)`
- `void killRunningDBusProcesses()`
- `QList<KRunner::QueryMatch> launchQuery(QString query)`
Important features – Deduplication

• Previously lots of duplicate results:
  • Recent documents and Baloo
  • Shell runner and applications
• Deduplication based on QueryMatch::id
• Explicitly enabled in metadata
All about the plugins - KRunner's ecosystem

- In KDE4 -> KDE5 transition, all scripted runners became unsupported
- With DBus-Runners, we once again have a thriving ecosystem!
- Various API improvements since initial creation
  - Lifecycle methods
  - Optimizations
  - Setting only specific actions for matches
  - RunnerSyntax support in metadata
All about the plugins - KRunner's ecosystem

• Promise to keep KF5 DBus-Runners compatible with KF6
• If you don't get a deprecation warning about the desktop file location ;)
• `${KDE_INSTALL_DATAROOTDIR}/krunner/dbusplugins` install location available since KF 5.72
All about the plugins - KRunner's ecosystem

• In Plasma 5.21 KDE Store integration was added
• Support for install scripts or prebuild deb/rpm/other packages
• Of course, warnings and recommendation for manual review implemented
• Terminal window was opened to allow manual intervention and adjustments
All about the plugins - KRunner's ecosystem

- Worked great and many more plugins on KDE Store
- But still not great UX
- Most install scripts for DBus-Runners are from official template
- Idea: Specify files in metadata format and install them without custom script
  - Could be extended with dependency installation/checks
Changes in KF6

• Lots of old code was removed and API refactored
• Most consumer-facing changes were prepared with deprecation macros
• Namespace was changed from "Plasma" to "KRunner"
• Significantly reduced dependency tree
Changes in KF6

• Model from Milou is now part of framework
  • Core functionality is in one place
  • More flexibility when changing code
• Kicker uses this unified model
• KRunner::Action as simple data class introduced
• Refactoring of threading was large effort
Changes in KF6 - Threading

- "Currently KRunner uses mental multi-threading for everything" - Kai
- Referring to AbstractRunner::match being called in different threads
- Prepare/teardown/init/reloadConfiguration are still done in main thread
Changes in KF6 - Threading

• Data initialization is a hassle if one does not want to block main thread
• Solution is to move runners to their own thread
• Prepare/teardown/init/reloadConfiguration are called in the runner's thread
  • Safe initialization and cleanup of data
• Even less code in frameworks
• Possibility for further optimizations
Further Plans & Ideas

• Overall, KRunner is in a pretty good state
• We utilize DBus-Runners where appropriate:
  • Plasma-Browser-Integration
  • KWin
  • Kactivitymanagerd
• Utility library for deduplicating C++ DBus-Runner code
Further Plans

• Sorting still needs improvements
• We use both match type and relevance for sorting
  • Match type is enum with magic values
  • If the match type is different, the higher one wins
  • Only if the match type is the same, relevance is considered
• Instead, separate sorting for categories and within category
• Smarter learning of your preferred results
Further Plans

- Making sort order configurable in plasmasearch KCM
- Fixed order for favorites
- Order within category unchanged
- Long requested feature with many duplicate reports
Big thanks to the community!

- All this would not have been possible without other members of KDE
  - David Edmundson
  - Natalie Clarius
  - Kai Uwe Broulik
  - Fushan Wen
  - Nate Graham
KRunner BoF!

- Tuesday 9:00 EEST in Room 2!
- Plan to discuss DBus-Runner ideas and visual improvements
- Hope to see you there!