When All Goes According To Script

Kevin Krammer, Software Engineer & Qt Trainer at KDAB

Qt WORLD SUMMIT 2015

Application Scripting

KDAB

- Non-binary extensions
  - Easily deployable
  - Separately shipable
  - Runtime (re-)loadable
- Automate tasks
  - Customer specific workflows
  - Macros
  - User defined script

Environments

KDAB

- Qt
  - QtScript
  - QJSEngine/QQmlEngine
- External
  - Lua
  - Python
  - ...

QQmlEngine

Wait! What?

KDAB

- I don't need any GUI
  - Fortunately QtQml does not have any UI dependency
- I don't need declarative programming
  - Only needs a boiler plate wrapper
  - But then you are missing out on the good stuff!

Basic Script Execution

KDAB

```
1  void ScriptRunner::runJavaScript(const QString &fileName,
2                                   const QString &mainFunction)
3  {
4      const QUrl fileUrl = QUrl::fromLocalFile(fileName);
5      const QString text = QString(QLatin1String(
6                  "import QtQml 2.2\n"
7                  "import "%1" as MyScript\n"
8                  "QtObject { property var result: MyScript.%2() }"))
9                .arg(fileUrl.toString(), mainFunction);
10
11     QQmlComponent component(m_engine);
12     component.setData(text.toLatin1(), fileUrl);
13
14     QObject *result = component.create();
15     if (component.isReady() && !component.isError()) {
16         qDebug() << result->property("result");
17         return;
18     }
```

Built-in Features

KDAB

- Includes
- Global "Qt" object
- Global "console" object for logging
- Easy access to data
- Easy access to objects
  - properties
  - signals
  - slots
  - invokable methods
- URL interception
- Custom global objects

KDAB

So, what am I missing out on?

Declarative Power

KDAB

- QtQml types ("Standard Library")
  - e.g. `Timer`
- Type from other QML Modules
  - e.g. `ListModel`, `XmlListModel`
  - or vendor/application specific modules
- Bindings
  - trigger evaluation on input data changes
- UI
  - e.g. QtQuick
- Easy instantiation of your own types
- Even define you own types!
  - e.g. for Mock objects

KDAB

Code or it didn't happen!

Summary

KDAB

- Skip `QJSEngine`, go directly for `QQmlEngine`
- Traditional scripting with a shim-loader
  - Already access to interesting built-in features
- Consider combining with Declarative programming

Any Questions?

More Cool Talks Today

KDAB

- 10:10
  - Effective Qt, Marc Mutz, right here
  - Qt on iOS, Mike Krus, B09
- 10:40
  - Among ELFs and DRARFs, Volker Krause, A05-06
- 11:30
  - Extending Qt on Android Apps using JNI, BogDan Vatra, B05-06
- 13:30
  - Integrating OpenGL with QtQtuick 2 Applications, Giuseppe D'Angelo, B09
  - Virtual Keyboard for Qt Applications, Tobias König, B05-06
  - Effective QML, Thomas McGuire, A03-04
- 14:30
  - Optimizing Qt Applications, Milian Wolff, right here
  - Qt Value Class Design, Marc Mutz, A05-06

Thank you!

www.kdab.com

kevin.krammer@kdab.com