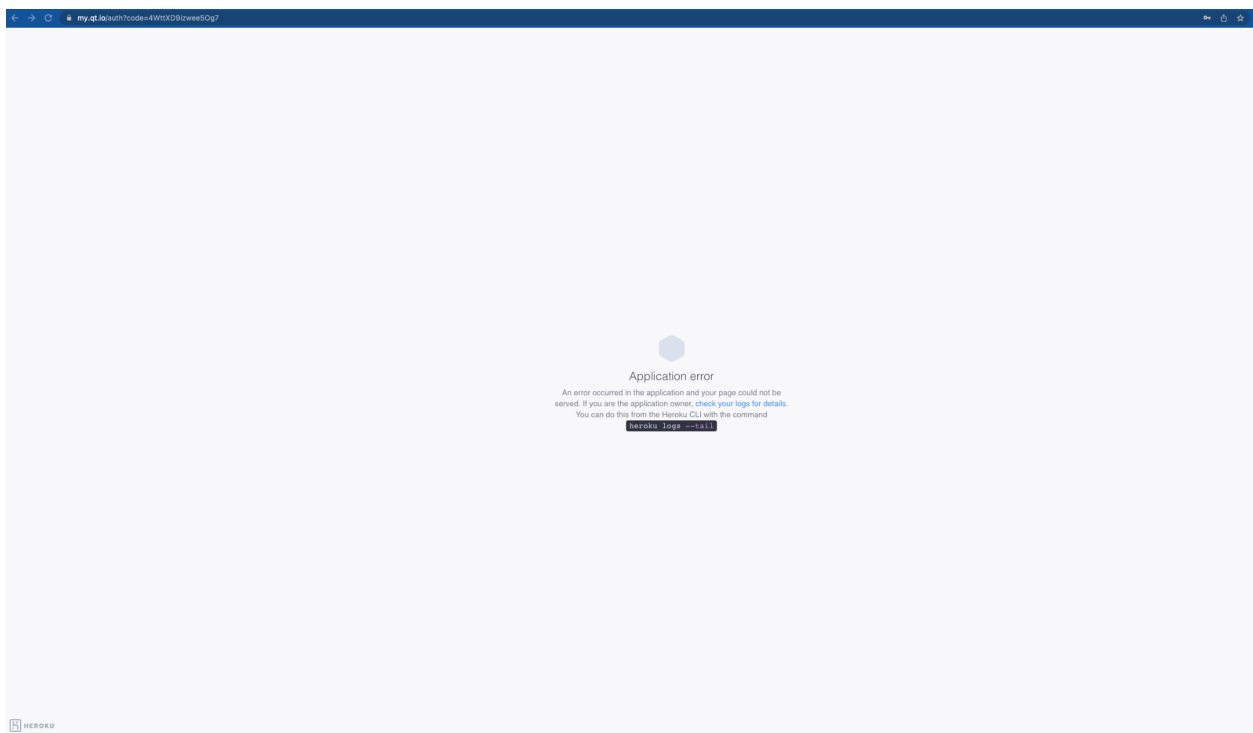


Used machine

My machine that I am using is a M1 Mac with Monterey 12.6

Getting Qt Creator installer

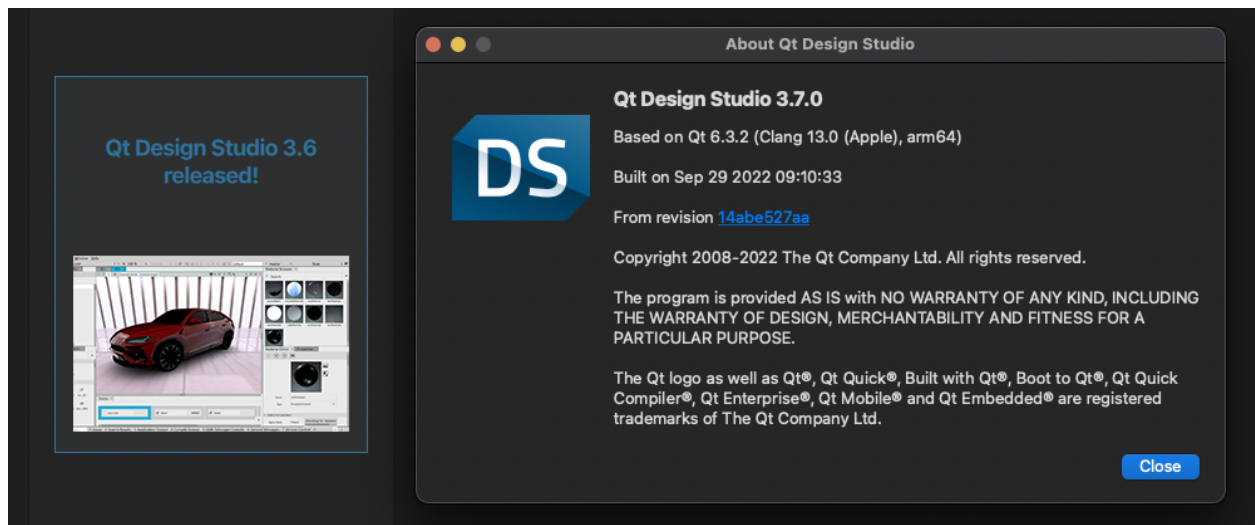
- The web site seemed a bit “too much”.
- I was searching for a “download Qt Creator” link, but all I could find was a link to the universal installer. At least for me it was a bit confusing.
- **In the installer it seemed a bit excessive to require an account for the installer to be able to install Qt via that.**
- Below is an error screen during the process of responding to the account verification email



- The installer process was a bit confusing as I thought that I had selected to install the Qt Creator, but it seems that I only installed the Qt Design Studio

First Use Experience

- Initial reaction for the first screen is a bit “much”. What I mean by that is it seems that it is more geared towards an entry level developer/ux designer.
- Personally really did not appreciate the web links at the bottom
- On the left side it says 3.6 released, which made me doubt that I had gotten an older version. Upon further investigation I actually had a newer version than what is advertised.



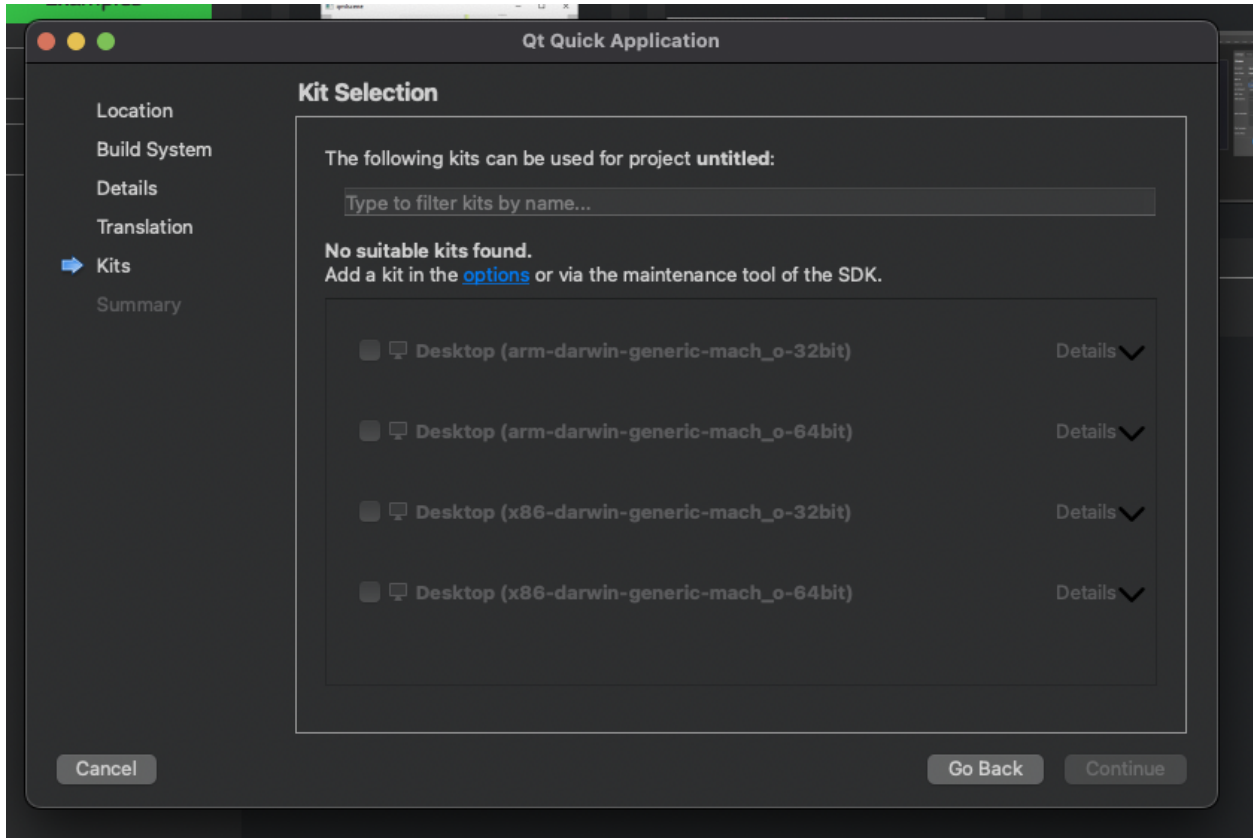
- The confusion w.r.t to the Studio vs. Creator installation was due to partial user error. Mac quick search did not initially find the Creator, thus leading me to launch the Studio.

The Creator first use experience

- Ok, my initial reaction was that the UI looks a bit “childish” and abundant. Just a bit too much.
- Luckily the quick actions for “Create Project” and “Open Project” options are available.
- But for example the most far left “action” column is IMO not really relevant here as you really do not have anything to debug for example.

Project Creation

- Plenty of different types ...started to think that are they really all needed, or just "left there" since they have always been there.
- What struck me is that for the "Application (Qt) type, the "Supported Platforms:" is empty... did my installer not install everything.
- **Also if the desired project folder does not exist, why not create it for the user**
- Creating a Quick application with default options seems to require a Kit ... dont recall Kits from my past Qt times.
 - I guess a Kit is a bundle of the build tools and dependencies.
 - Ok, so apparently my installation has some Kits, but the UI does not tell me what is wrong with my Kits.
 - After a while I determined that for some reason I did not install any Qt version during the initial process
 - Why the installer did not force me to install a version of Qt, as how can I develop apps for Qt without a version of Qt?
 - Also it took me some googling to find out about the Maintenance Tool, wonder nothing pointed me to that tool in Creator
 - **What should preferable happen is that the Kit prefs could show a dialog of available Qt version and once I select a version it would automatically install it.**
 - Hooray ... after a while waiting for the installation to complete I finally got my qml test app running!!

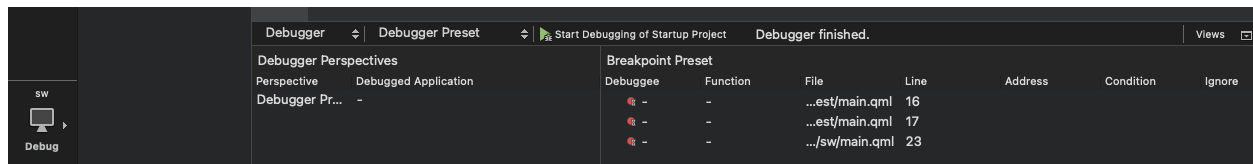


Simple observations

- Initially thought that no Hot Reloading was available as the QML UI did not update on save. But after discovering QML Preview from the Build menu it seems to be at least partially working
- Example Apps (Coffee) not automatically configured when previously no Kits were available
- Simple find for docs/files/functions => **Universal search**
- For some reason I would like to see directory structures in the Project view
 - And to add to this a bit more ... the projects view started to show “generated” files and targets, that IMO are build outputs and as such are cluttering up the view
- When looking at docs, for example, <https://doc.qt.io/qt-6/cmake-manual.html>. It would be great if there is a github link or similar the the actual full code

Debugging

- When having multiple files open the CMR+R command runs the active project, which was a bit confusing
- **Stopped breakpoint hovering...i.e. On top of objects / variables, tooltip etc.**
- **The QML debugging is quite finicky. A wizard based project can be debugged quite easily. But when I created a seemingly same kind of a project from scratch I was unable to debug QML with it.**
- **Automatic indentation removes breakpoints even if the actual line contents is still valid**



- **The debugger somehow manages to use breakpoints from another project. Only BP #23 should be active, #16 and #17 are from another project**

Editing QML

- Autocomplete works a bit "late". When you are learning or are unfamiliar with QML, the auto-complete comes up only after three letters of an existing property.
 - *NOTE: There's a setting for this, but perhaps consider revising the defaults :)*
- It seems that sometimes the editor shows errors and sometimes not, for example assigning a ColorAnimation to a color property is not highlighted in the editor as an error
- **Auto indents does not work ... have set it to 2 spaces, but 4 exists**
 - **Ah, there is a setting, but the current way of setting it is a bit confusing as I only changed *Text Editor* settings**
- **Adding a new file under a directory that does not exists should suggest to create the dir**
- .qml file does not report an error always when the item is not found

```
GraphQLConnection {
    id: gql
}
}
```

- But runtime error occurs
13:31:43: Starting
/Users/mikko/src/qt-analysis/build-star-wars-qml-Desktop_arm_darwin_generic_mach_o_64bit-Debug/starwarsqml...
QQmlApplicationEngine failed to load component
qrc:/qml/main.qml:35:5: GraphQLConnection is not a type

- A nice feature would be to be able to wrap an existing qml item with a new one, from the gui.
- Once I got a few settings tweaked the editing feels snappy and responsive

Navigation, i.e. moving between files

- In the case of shadow builds it would be great is the build output is more easily navigable in Creator
- Also the File System navigator is IMO too broad. Roughly speaking the only thing I want to navigate is my sources and build outputs.
- Now after a few days of usage the “navigation tree” it seems fairly useless at least to me
- Loving the Command+K function!

Version Control

- CVS ... chuckles ;)
- When adding git (git init) Creator does not automatically detect it ... also when “adding it via the GUI” it felt like creating a new repo, even though it already existed
- **Assumed that diff against latest HEAD would be automatically visible in editor**

```

13 console.log(e)
14
15 var response = gql.query("
16 query Query {allFilms {
17   films {
18     id
19     [Skipped 1 lines...] title
20   }
21 }
22 })
23 console.log(response)
24
25 ListModel {
26   id: films
27   onColumnsInserted: () => console.log("INSss")
28 }
29
30 MouseArea {
31   id: m

```

- The diff view is quite contrasty. To me this is quite un-usable. While I do believe that offering a lot of adjustability and configurability is a good thing. The defaults could be perhaps a bit better.
- **Also the diff opens in a separate “view” which imo is unnecessary and hinders its usability. Maybe consider some sort of folding out view per changed blocks/lines**
- Previous diff files do not update themselves when the diff changes.

- Commit chunk staging is good

General Editor Observations

- If the underlying directory is moved/renamed the editor thinks that it is editing a read only file
- Still after a week's worth of using it the UI feels messy (yeah, it's kind of a strong word).
 - There are a lot of dock panels
 - It feels that in order to accomplish a task you always need to switch to a different view
- The performance of Qt creator seems to be snappy. My test project is quite small, so I might not be able to evaluate this to the max, but so far all is good
- Perhaps revisit some of the default theme colors and some are very good, but other have too much red, used for example properties. But again a more of a personal opinion.
-

Refactoring

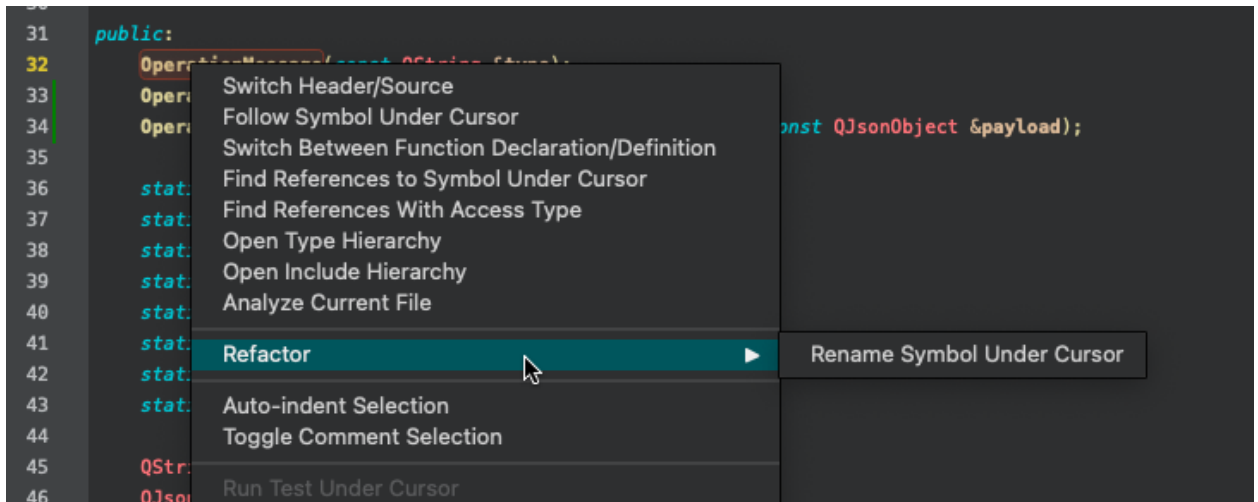
- **Generating a constructor allows duplication of existing constructors. This could be detected prior generation**

```
34 OperationMessage(const QString &id, const QString &type, const QJsonObject &payload);  Previous declaration is here
35 OperationMessage(const QString &id, const QString &type, const QJsonObject &payload);  Constructor cannot be redeclared
```

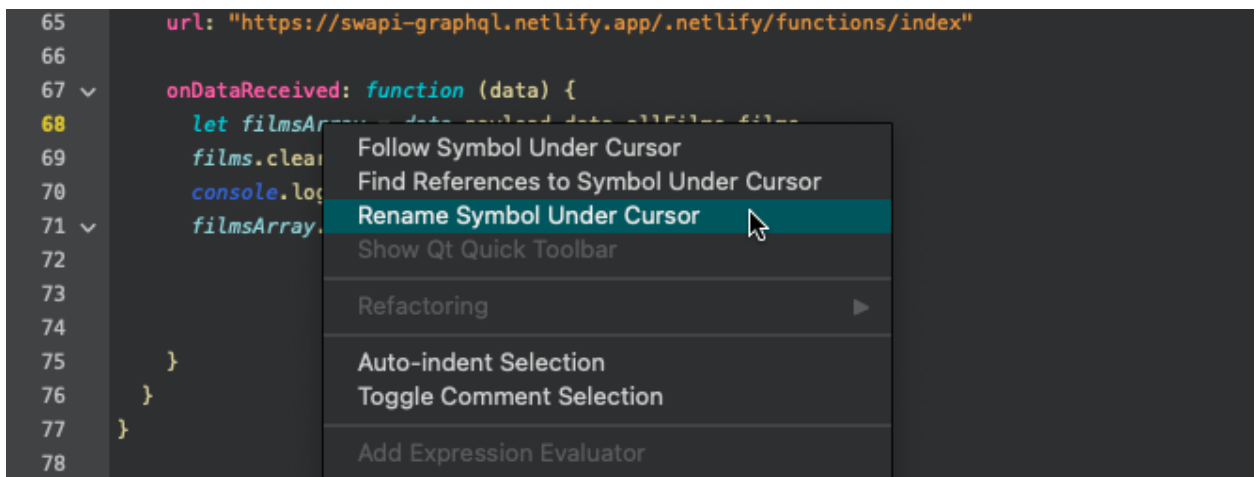
- Consider adding a “Quick-fix” option to warnings that could be easily automatically fixed by the IDE

```
54 void stateChanged(ConnectionState state);  signal arguments need to be fully-qualified (GraphQLWebSocketConnection::ConnectionState instead...
53 Q_PROPERTY(bool isConnected READ isConnected)  Q_PROPERTY should have either NOTIFY or CONSTANT [clazy-qproperty-without-notify]
```

- There is a difference between QML and C++ files when renaming a symbol under cursor. In C++ it is under the Refactor menu, consistency would be good.



But in QML files it is a top level function



- Directory refactoring is not possible
- After adding a function definition to a header file manually it would be great if the boiler plate implementation function would be automatically added, or at least a prompt or a similar to ask if to generate it.
 - Yes, there is a menu option for this.
- **Overloaded class constructor name refactor does not change the class name only the specific constructor**
- Also it seems that the qml refactoring is more or less limited to variable name changes, perhaps in the long run this needs to develop more.

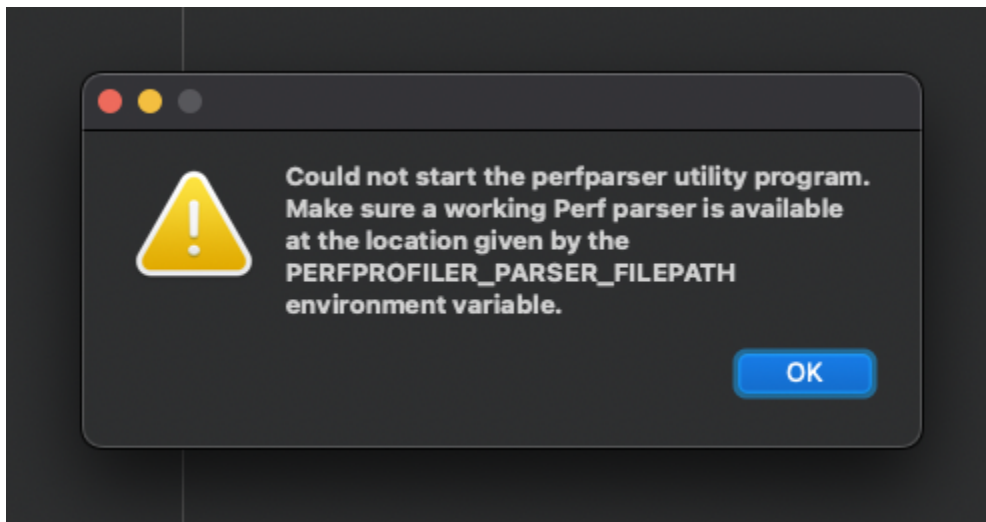
Mobile device targets

- My installation suffered from this bug <https://bugreports.qt.io/browse/QTBUG-107710>
 - Manually installed a newer version of CMake -> FIX

- I was pleasantly surprised at the fact that how easy it was for me to get a simple QML app running on the iOS simulator, granted I already had all necessary tooling installed on my mac, but still!!
- I could not get my cmake scripts to install a dynamic library QML plugin library that was part of the project.

Performance

- Could not start the performance analyzer. Consider adding the tool as part of the default installation.



- Also when searching for it from the Maintenance Tool I could not find it. Not quite sure then what this tool refers to. Is it this one <https://github.com/qt-creator/perfparsers>?
- Is it possible to do on mac?
 - If not consider adding a mention to the user that it is only available on a specific platform

Summary of Qt creator - “my gut feeling”

I set on this project to try and see if I would use Creator as my editor if I were to end up in a project that had to do with QML and Qt. The answer at the time of writing this is no, I would not. Let me try to summarize my experience.

I constantly had an overall feeling of slight “messyness” of the views. I easily ended up in situations where the UI had half a dozen different docked views open. Also often I found myself looking for functionality that I had already used i.e. “where did I see that configuration option”. I did not get a good feel on navigating between files. I found myself ping-ponging between the file

system view and the project view. In my vantage point there should be only one view. I did like the "Locate" functionality.