

Einführung in QT4

QT 4 Einführung:

QT ist ein Toolkit von der Firma [Trolltech](#), für Plattformübergreifende GUI Anwendungen.

Wer sich darunter noch immer nichts vorstellen kann dem empfehle ich dem deutschen [Wikipedia](#) Eintrag über QT.

Diese Einführung soll einen kurzen und simplen Einblick in diese Bibliothek zeigen.

Für eine sehr Ausführliche Dokumentation kann ich nur an auf die [Trolltech Dokumentation](#) verweisen.

Diese Seite sollte für alle QT Programmierer als Referenz dienen, sind aber in Englisch und für eher fortgeschrittene Programmierer gemacht.

Das Klassische Hallo Welt mit QT4:

Quellcode

```
1. // Hier werden die benötigten Header Dateien geladen
2. #include <QApplication>
3. #include <QPushButton>
4. int main(int argc, char *argv[])
5. {
6. // Hier wird das QT_Programm erstellt
7. QApplication app(argc, argv);
8. // Hier wird ein Button erstellt der dann nachher das Programm beenden kann
9. QPushButton beenden("Beenden");
10. // Hier wird eine Größe für den Knopf festgelegt,
11. // ohne diese Zeile wäre der Knopf so klein
12. // wie es der Text im Button zulassen würde.
13. // (Breite / Höhe)
14. beenden.resize(200, 30);
15. // Hier wird das SIGNAL "clicked" vom Button "beenden",
16. // dem SLOT "quit" vom Hauptprogramm "app" übergeben.
17. QObject::connect(&beenden, SIGNAL(clicked()), &app, SLOT(quit()));
18. // Hier wird der Knopf beenden im Programm noch sichtbar gemacht.
19. beenden.show();
20. // Startet das Programm
21. return app.exec();
22. }
```

Alles anzeigen

Das Signal und Slots Konzept ist eines der Hauptmerkmale von QT.

Dieses Konzept wird an einer anderen Stelle in naher Zukunft genauer erklärt.

Zum Übersetzen des Quellcode verwendet man am einfachsten das Tool " qmake".

qmake ist ein kleines Programm das Teil von QT ist und mit diesem Programm kann man zuerst einmal einen Projektdatei automatisch erstellen lassen. Und mit dieser Projektdatei lässt sich dann wieder mit qmake ein Makefile erstellen. Und mit diesen Makefile wird dann das Programm erstellt, aber hier einmal der Übersetzenvorgang noch einmal der Reihe nach.

Zuerst wird mittels

Quellcode

1. qmake -project -o hallowelt.pro

Eine sogenannte Projektdatei erstellt, in dieser Datei sind Informationen die für die Makefile Erstellung notwendig sind (Wie z.B.: Linkerdaten , usw), aber für den Anfang ist es nicht nötig hier irgendetwas zu ändern.

Danach sollte eine Datei "hallowelt.pro" im Ordner liegen.

Jetzt reicht der Aufruf von

Quellcode

1. qmake

um eine "Makefile" Datei erstellen zu lassen.

Und mit Hilfe dieser Datei lässt der Aufruf von,

Quellcode

1. make

wenn wir alles richtig gemacht haben, die ausführbare Datei "hallowelt" erstellen.

Die hier vorgestellt Übersetzungsprozedur ist auf Linux zugeschnitten.

Projekt Datei:

In der Projektdatei werden die Informationen für "qmake" zu Verfügung gestellt.

der Aufbau ist wie folgt:

Quellcode

1. # Hier kann man zB mit "release" Optimierungen einschalten oder
2. # mit "debug" qt-Debuginformationen hinzufügen oder
3. # Compilervorwarnungen aus "warn_off" und einschalten "warn_on"
4. CONFIG += debug
5. # Was erstellt werden soll zB:
8. # "lib" für eine Bibliothek
9. # "app" für ein Programm
10. TEMPLATE = app
12. # Wie das Programm heißen soll
13. #(ansonten verwendet qmake denn Ordnernamen als Programmnamen)
14. TARGET = hallowelt
16. # Ordner mit Quell- oder Headerdateien
17. DEPENDPATH += ./src
18. INCLUDEPATH += ./src
20. #Wo die Objekt- und Mocdateien abgelegt werden sollen
21. OBJECTS_DIR = ./tmp
22. MOC_DIR = ./tmp
23. # Liste aller Dateien des Projekts
25. HEADERS += alleHeaderdateien.h
26. SOURCES += alle.cpp quellcodedateien.cpp

Alles anzeigen

Bei kleineren Projekten ist die Automatischegenerierung der Projektdatei meist ausreichend.

<http://www.easy-coding.de/wiki/Entry/24-Einf%C3%BChrung-in-QT4/?s=076403ffd298b182ace67573fcbd0c4db4b4abb1>

Es kann aber sinnvoll sein zB. die Header Dateien in deinen Unterordner zu packen, und dann findet qmake bei der Automatischengenerierung diese Dateien nicht mehr. Ab hier ist eine eigene Erstellung der Projektdatei notwendig.

Eine genaue Erklärung der einzelnen Funktionen ist auf der [Trolltech](#) Seite zu finden(Englisch).