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 Programmiere gemacht.

Das Klassiche Hallo Welt mit QT4:

Quellcode

```
// Hier werden die benötigten Header Datein geladen
#include <QApplication>
#include <QPushButton>
int main(int argc, char *argv[])
{
// Hier wird das QT_Programm erstellt
QApplication app(argc, argv);
// Hier wird ein Button erstellt der dann nachher das Programm beenden kann
QPushButton beenden("Beenden");
// Hier wird eine Größe für den Knopf festgelegt,
// ohne diese Zeile wäre der Knopf so klein
// wie es der Text im Button zulassen würde.
// (Breite / Höhe)
beenden.resize(200, 30);
// Hier wird das SIGNAL "clicked" vom Button "beenden",
// dem SLOT "quit" vom Hauptprogramm "app" übergeben.
QObject::connect(&beenden, SIGNAL(clicked()), &app, SLOT(quit()));
// Hier wird der Knopf beenden im Programm noch sichtbar gemacht.
beenden.show();
// Startet das Programm
return app.exec();
}
```

Alles anzeigen

Das Signal und Slots Konzept ist eines der Hauptmerkmale von QT.
Dieses Koncept 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 sogenante 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. # Compielerwarungen aus "warn_off" und einschalten "warn_on"
4. CONFIG += debug
5.
6. # Was erstellt werden soll zB:
7. # "lib" für eine Bibliothek
8. # "app" für ein Programm
9. TEMPLATE = app
10.
11. # Wie das Programm heißen soll
12. #(ansonten verwendet qmake denn Ordnernamen als Programmnamen )
13. TARGET = hallowelt
14.
15. # Ordner mit Quell- oder Headerdatein
16. DEPENDPATH += ./src
17. INCLUDEPATH += ./src
18.
19. #Wo die Objekt- und Mocdatein abgelegt werden sollen
20. OBJECTS_DIR = ./tmp
21. MOC_DIR = ./tmp
22.
23. # Liste aller Datein des Projekts
24. HEADERS += alleHaederdatein.h
25. SOURCES += alle.cpp quellcodedatein.cpp
```

Alles anzeigen

Bei kleineren Projekten ist die Automatischegenerierung der Projektdatei meist ausreichend.

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 einen eigene Erstellung der Projektdatei notwendig.

Eine Genau Erklärung der einzelnen Funktionen ist auf der Trolltech Seite zu finden(Englisch).