Archived:Simple Helloworld in Qt

Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

Qt Quick should be used for all UI development on mobile devices. The approach described in this article (based on QWidget) is deprecated.

In this (archived) article we will learn how to start Qt programing using Qt in Carbide.c++. We will create a simple **Hello World** program which can be written in Qt.

Note: This article is based on **Carbide C++** and **Qt SDK for Symbian** (which may be deprecated later), if you would like to use **Qt Creator** with new Nokia Qt SDK then please don't follow this article. It is better to use Nokia Qt SDK instead.

Prerequisites

Note :If you already have Carbide C++ setup and you are the Symbian developer then move on and just install **Qt SDK** only, Or else follow this steps

1. Install Active Perl
2. Install JRE
3. Install S60 SDK
4. Install Carbide C++
5. Guide to install Carbide C++ (Optional)
6. Install Qt If you don't find it please try here

Creating Hello World Project

1. Launch Carbide C++
2. Click on **File->New->Qt Project->Select Qt GUI Main Window (Under Qt GUI) -> Put project name ->Select SDK->Check Qt Modules**
3. Your project will be created and ready to build.
4. Open **main.cpp** from the project explorer
5. Paste the below code in your **main.cpp** file. Build and run the project.

```
#include <QApplication>
#include <QPushButton>
int main(int argc, char *argv[])
{
QApplication app(argc, argv);
QPushButton helloButton("Hello World");
helloButton.resize(80, 20);
helloButton.show();
return app.exec();
}
```

First we include the definitions of the `QApplication` and `QPushButton` classes. For every Qt class, there is a header file with the same name (and capitalization) as the class that contains the class's definition.Inside {{Icode|main()} the `QApplication` object to manage application-wide resources. The `QApplication` constructor requires **argc** and **argv** because Qt supports a few command-line arguments of its own. The `QPushButton` which is a button widget class in Qt and we set the button name as **Hello World**. `resize()` helps us to make the size of the button. `show()` display the button on the screen.

Screenshot

This screenshot was taken from Qt running on an old Symbian emulator (S60 3rd Edition FP2).