ICS

Qt For Beginners Training

Part 1

Where to Download Qt?

- Commercial and open source licenses available
- https://www.qt.io/download (Online Installer)
- https://download.qt.io/official_releases/qt/
  - (Source and instructions on how to build)

- Many Linux distributions provide Qt as a native package
- Be aware of license requirements

What is Qt?

- Cross-Platform Software Toolkit and API that allows to develop World-Class Desktop and Embedded Devices applications
- Spelled Qt and not QT
- Pronounced “cute”
- Is owned by **The Qt Company**
- **The Qt Company** is responsible for all Qt activities including product development
- https://resources.qt.io/customer-stories-all

Companies Using Qt

Where to Get Help?

- Documentation in Qt Assistant or Qt Creator
- Qt's examples: $QTSRC/Examples
- Qt Project: http://www.qt.io/developers/
- Qt Centre Forum: http://www.qtcentre.org/
- Mailing lists: http://lists.qt-project.org
- IRC: irc.freenode.org channel: #qt
- Bug Report: https://bugreports.qt.io
- Tutorials: http://wiki.qt.io/Developer-Guides
- Video tutorials: www.youtube.com/user/QtStudios

Widgets vs Qt Quick (QML)

- Two ways of writing GUI
- The QWidget API is in C++, compiled and more suitable for Desktop Applications
- QML is a markup language which uses Javascript and is interpreted at run -time. More suitable for embedded devices.

Starting Your Project with Widgets

```
#include <QtWidgets>

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    QPushButton button("Hello world");    // Creates a widget
    button.show();
    return app.exec();                    // Starts the event loop
}
```


Program consists of:

- main.cpp – application code
- helloworld.pro or CMakeLists.txt - project file

Starting Your Project with Qt Quick

Hello world

```
#include <QGuiApplication>
#include <QQmlApplicationEngine>

int main(int argc, char *argv[])
{
    QGuiApplication app(argc, argv);
    QQmlApplicationEngine engine;
    engine.load(QUrl(QStringLiteral("qrc:/main.qml")));


    return app.exec();
}
```

Qt Quick - Hello World in Qt

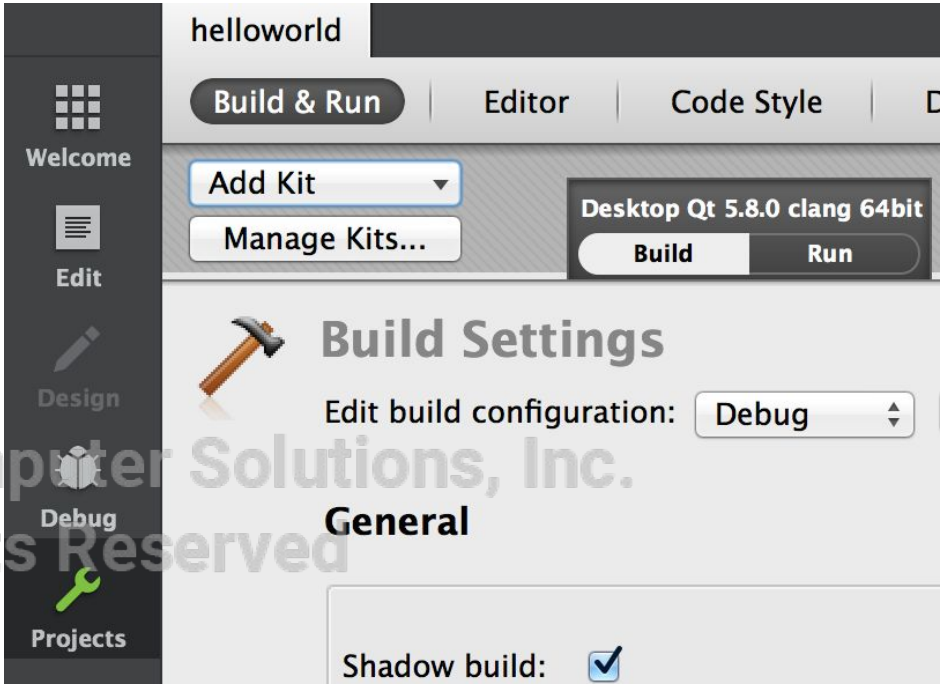
```
import QtQuick 2.4
import QtQuick.Window 2.2

Window {
    visible: true
    width: 360; height: 360
    Rectangle {
        anchors.fill: parent
        MouseArea {
            anchors.fill: parent
            onClicked: {
                Qt.quit()
            }
        }
        Text {
            anchors.centerIn: parent
            text: "Hello World"
        }
    }
}
```

Kits

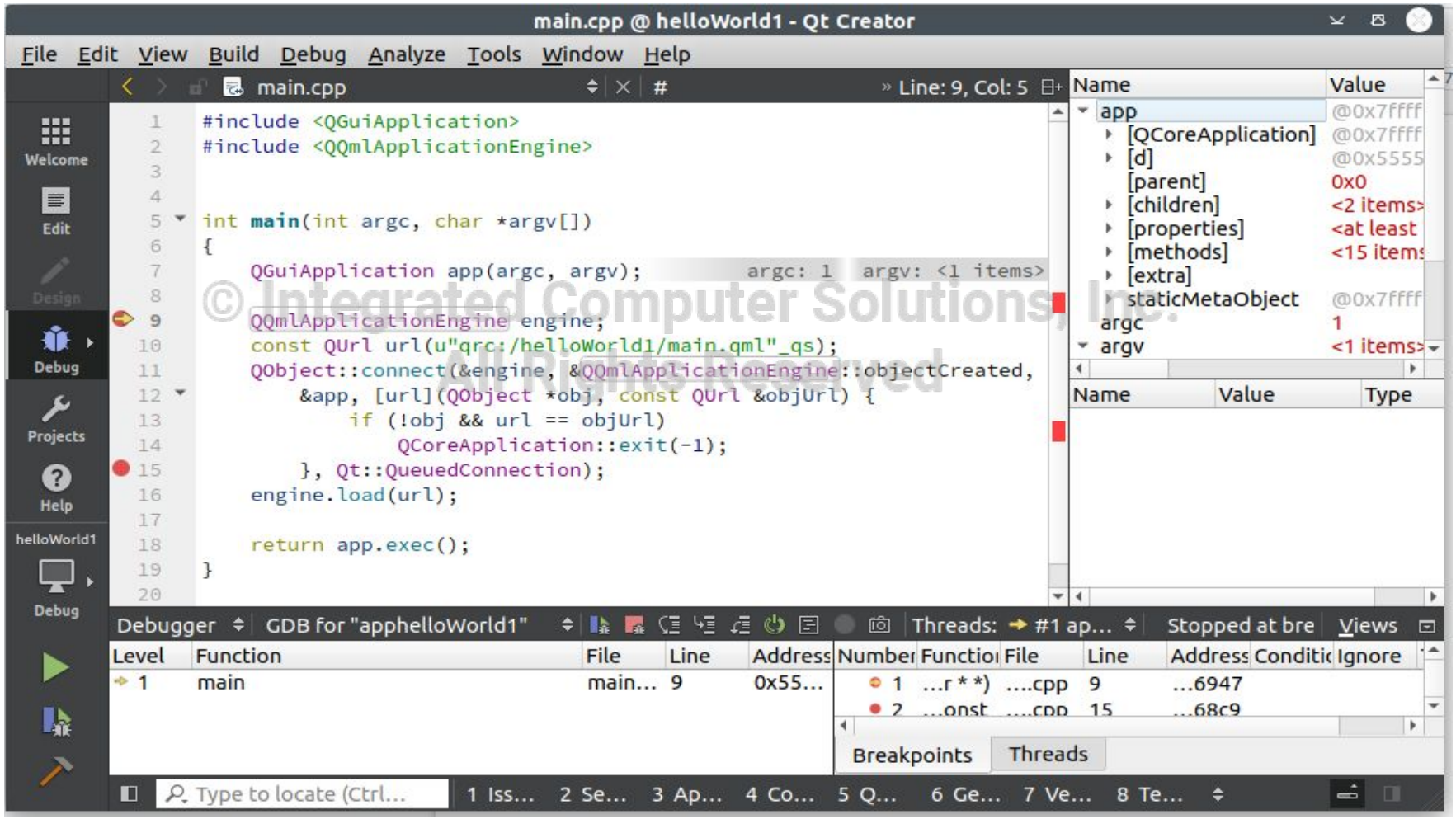
A kit defines, which

- Qt version is used (Qt versions sheet – qmake location)
- compiler is used (Compilers sheet)
- debugger is used (Debuggers sheet)



Debug Screen

Debug – > Start Debugging (or F5)



Qt Creator Shortcut Keys

Locator Prefixes

- F2 - Go to a symbol definition
- l <line number> - Go to a line in the current document
- ? <help topic> - Go to a help topic
- o <open document> - Go to an opened document
- https://www.kdab.com/development-resources/qtcreator/

Qt Books

- C++ GUI Programming with Qt 4 (2nd Edition) (Prentice Hall Open Source Software Development Series), Jasmin Blanchette and Mark Summerfield
- Practical Qt: Real World Solutions to Real World Problems, Matthias Kalle Dalheimer and Jesper Pederson
- An Introduction to Design Patterns in C++ with Qt 4, Alan Ezust and Paul Ezust

In the sessions

This five-part training will also cover:

- QVariant, QObject
- QML
- QWidgets
- Model/View