# Mobile Phone Programming

Quick start: Qt for Windows/Linux, Symbian and Maemo/Megoo

## Getting Qt: Available Qt SDKs



Stand-Alone Platform SDK:



#### Download: http://qt.nokia.com/downloads

## Qt SDK Content

- All SDKs contain:
  - Qt Source / libraries



- Qt Creator Qt development IDE
- Qt Documentation API documentation, examples etc.
- Qt Designer Drag and drop GUI creation
- Qt Linguist i18n (translation tool).
- Nokia Qt SDK
  - Qt Simulator testing your mobile applications on the desktop.
  - Qt Mobility APIs for mobile development

## Nokia Qt SDK (version 1.0)

#### Versions available

- 32- or 64-bit Microsoft Windows XP Service Pack 2, Windows Vista, or Windows 7.
- 32-bit Ubuntu Linux 7.04 or later.
- 64-bit Ubuntu Linux 7.04 or later.
- Apple Mac OS X 10.6 or later (Beta).

#### Notes

- Linux and Mac support Maemo only
  - Remote Compiler can be used (beta).
- General on Windows
  - Install all tools to the same drive (e.g., C:\)
  - $\circ\,$  Do no use network drives.
  - Use default installation paths. Avoid special characters or spaces in paths.
- Maemo do not use special characters in the project name



### Running "hello world"

- Lets get started by creating and running a "hello world" on the desktop, in the Qt simulator, on Symbian and Maemo
- 1. Download and install Nokia Qt SDK
- 2. Download and install the appropriate desktop SDK
- 3. Launch Qt Creator
- 4. Create a new project
- 5. Add source files
- 6. Compile and run on the target devices

### Installation:

- Download preferred version for your platform
  - Offline to download the entire SDK in one package (update later).
  - $\circ$  Online to let installer get latest version.
- For Windows, run setup.
- For Linux
  - o chmod u+x installer\_filename
  - o ./installer\_filename



- File -> New Fil **Project**
- Empty Qt proj
- Set project na location

lew File or		Choose a template:	
		Referts	C++ Library
		C++ Project	Qt Unit Test
	1	Other Project	C Qt Custom Designer
		Project from Version Control	Empty Qt Project
)t project		Files and Classes	import Existing Project
(i projeci		C++	
		Qt	
		General	
ect name an	nd		Creates a gmake-based project without any files. This allows you to create an application without any default classes.
	mpty Qt Proje	ect	? ×
	> Location	Introduction and p	roject location
	Qt Versions		Ord evelopet. Add files to it later on humaine the
	Summary	other wizard generates an empty	QC4 project. Add riles to it later on by using the
	Dammar y		
		Name: helloworld	
		Create in: C:\qtprojects	Browse
		🔲 Use as default project loca	ation
$\frown$			
			Next > Cancel

 Select the targets you want to support

🗖 Empty Qt Projec	t
Location	Qt Creator can set up the following targets for project <b>helloworld</b> :
Summary	Desktop
	Maemo     Qt for Fremantle PR1.2 Devices (Nokia Qt SDK)     Qt Simulator     Simulator     Simulator Ot for MinGW 4.4 (Nokia Ot SDK)
	<ul> <li>Grand Symbian Device</li> <li>Qt for Symbian Devices (Nokia Qt SDK)</li> </ul>
	< <u>Back</u> <u>N</u> ext > Cancel

 Finish the project creation

	🗮 Empty Qt Projec	t	? ×
	Location Qt Versions	Project management   Add to project:   Add to version control:   Image: C: \qtprojects\helloworld:	*
4		< <u>B</u> ack <u>F</u> inish Car	ncel

- Right click project -> Add New
- Select C++ source file.
  - Name: main.cpp
  - Path: project path (default)
- Accept default project management options and finish.

	Projects	▾◲◓∃×◂◗		
٩dd	🔂 hellowo	Build Project "helloworld" Rebuild Project "helloworld" Clean Project "helloworld" Run gmake in C:\gtprojects\belloworld-build-desktop		
		Run		
		Close Project "helloworld"		
).		Add New Add Existing Files		
🕵 New C-	++ Source File	×		
Location Choose the Location				
Summary Name: main.cpp				
	Path: C:\q	tprojects\helloworld Browse		
	New C++ Source	File		
	Location	Project management		
	Summary	Add to project:		
		Add to version control: None>		
		Files to be added in		
		C:\qtprojects\helloworld:		
		< <u>B</u> ack <u>F</u> inish Cancel		

• Adding our "hello world" source code to the empty main.cpp

#include <QtGui>

```
int main(int argc, char *argv[])
```

```
QApplication app(argc, argv);
QLabel *label = new QLabel("hello world");
label->show();
```

return app.exec();



### Hello World - Elements



- QApplication
  - $\circ$  One per GUI application
  - $\circ$  Manages application wide resources.
  - Contains the application event loop (handles mouse clicks, etc.).
- QLabel
  - Qt GUI element (called a Widget)
  - $\circ$  Useful for displaying text and images

```
#include <QtGui>
```

```
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    QLabel *label = new QLabel("hello world");
    label->show();
    return app.exec();
}
```

## Nokia Qt SDK: Simulator

- Efficient Testing
  - Quick Launch
  - $\ensuremath{\circ}$  Scripting possibilities
  - Simulate phone events & Qt mobility events
  - Skins for different platforms
  - $\circ$  Not a full phone emulator.

📱 🖿 💿 JS			-1
imulate			. 1
Generic		¢	<u>}</u> ^
Storage			
Network			
Location			
Contacts			
Messaging			
Sensors			
Scripting			
			11
Application		_	Ŧ
<ul> <li>Application</li> <li>✓ View</li> </ul>	_	_	÷
Application     View svice Symbian Touch	_		, ¢ ,



## Launching Qt Simulator

Run "hello world" in simulator
 Select Qt Simulator target
 Click play

NOKIA		N9C0	
	Simulator	×	1
hello world			



## Running on Maemo

- Two development options
  - Full Linux Environment (Scratchbox)
    - Guide: http://wiki.maemo.
      - org/Documentation/Maemo\_5\_Final\_SDK\_Installatio

n

- Using a Virtual Image: http://maemovmware.garage. maemo.org/2nd\_edition/
- Nokia Qt SDK (based on MADDE)
  - MADDE stands for Maemo Application Development and Debugging Environment.
  - Allows on device debugging from within Qt Creator
  - Works on Windows, Linux, Mac OS X (coming soon).

We use the Nokia Qt SDK - howto can be found here: http://bit.ly/b29q3h

#### Run "hello world" on the N900

- Select the Maemo target
- Press the run button



## Running on Symbian

- Building Symbian applications currently require Windows
   Or use of Remote Compiler (not covered here)
- Depending on the device OS we have to install some prerequisites:



- *path*\NokiaQtSDK\Symbian\sis\qt\_installer.sis Qt libraries *path*\NokiaQtSDK\Symbian\sis\qtmobility.sis Qt Mobility
- Install by sending files over Bluetooth or use Nokia PC Suite or Ovi Suite

## Automatic Deployment & On-Device Debugging

- Requires debug agent running on device: App Trk
- Install:
  - *path*\NokiaQtSdk\Symbian\sis\s60\_5\_0\_app\_trk\_3\_x\_x.
     sisx
  - Install to phone memory (C:\)
  - $\circ$  Also required on Symbian^3 devices
- Start the TRK application on the device
  - $\circ$  Create connection via Bluetooth or USB
  - $\circ$  **Recommended**: USB

Howto can be found here: http://bit.ly/d9Y3OA

### Using Qt Creator



- http://developer.qt.nokia.com
- http://qt.nokia.com/doc





- Install the SDK on you computer
- Create and run "hello world".
- Try some examples.