
# Qt for Symbian

**Liu XiaoGuo (Liuxg)**

**Forum Nokia, China**

NOKIA


# Agenda

- Introduction to Qt
- Qt development on Symbian
- Smart installer
- Qt Mobility project
- Demos

*Have you ever cried when developing mobile apps?*

*It is such a pain!*


# Qt: A brief introduction

## Founded in 1994

- Trolltech acquired by Nokia in 2008
- More than 250 employees in eight locations worldwide
- Trusted by over 5,000 customers worldwide

## Qt: a cross-platform application and UI development framework

- For desktop, web and embedded development
- Used by more than 250,000 commercial and open source developers
- Backed by Qt consulting, support and training

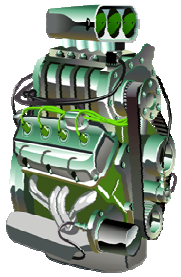
NOKIA

# Why Qt?

## What is Qt about?

- **QuickTime player? Programming language? A platform? Yet another runtime?**
- **A new Japanese car model?**

## Qt is Cross-platform application framework



- Differentiated user experience across hardware platforms
- **Hybrid development**: convergence of web and native applications
- Cross-platform software **across desktops and mobile devices.** Qt supports major desktop OSs already and will extend it's reach to Nokia platforms.
- Qt is **not a runtime** – no perf issues or sandbox restrictions. You can stand out with your native development skills to fulfill the use cases offered by Qt.
- Qt Mobility APIs will bring in the easy to use APIs for mobile use cases.

## Shorter time to market

- Productive APIs and comprehensive documentation with great examples
- Delivering functionality faster and maximizing efficiency by placing focus on innovation
- All the great tools offered free of charge - commercial services available as well

NOKIA

# Qt is used everywhere

## From embedded devices to desktop applications

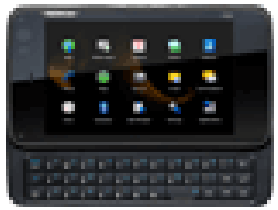
## By companies from many industries

AMD
Google
Canon
Alcatel·Lucent
BT
DAIMLER
EPSON
esa
freescale semiconductor
HITACHI Inspire the Next
GENERAL DYNAMICS C4 Systems
Mentor Graphics
PGS
HONDA The Power of Dreams
MITSUBISHI HEAVY INDUSTRIES, LTD.
NORTEL NETWORKS
ORACLE
PHILIPS
Pioneer
skype
Raytheon
Schlumberger
ROHDE&SCHWARZ
SCANIA
SENNHEISER
SHARP
xerox
VOLVO
vodafone
THOMSON REUTERS

NOKIA

# Qt Facts

http://qt.nokia.com



- Qt is easy and intuitive to learn
- Qt documentation is comprehensive with good examples
- Qt supports major desktop Operating Systems already and has extended its reach to Symbian and Maemo
- Qt is not a runtime – no performance issues, no sandbox restrictions
- **Most importantly**, Qt will be the main choice for developers targeting Nokia platforms and Symbian^4

  http://developer.symbian.org/wiki/index.php/Symbian%5E3

NOKIA


# Qt licenses support all business models

| | Commercial | LGPL v. 2.1 | GPL v. 3 |
|---|---|---|---|
| **License Cost** | License fee charged | No cost | No cost |
| **Must provide source code for changes to Qt** | No, modifications can be closed | Source code must be provided | Source code must be provided |
| **Can create proprietary application** | Yes—no obligation to disclose source code | Yes, if dynamically linked to Qt library | No, application is subject to the GPL |
| **Support** | Yes, with valid maintenance agreement | Not included, available separately | Not included, available separately |
| **Charge for Runtimes** | Yes—in some instances* | No, distribution is royalty free | No, distribution is royalty free |

• Runtime charges apply when the Qt-based application is part of a joint hardware and software distribution and the main UI of the device is controlled by Qt.

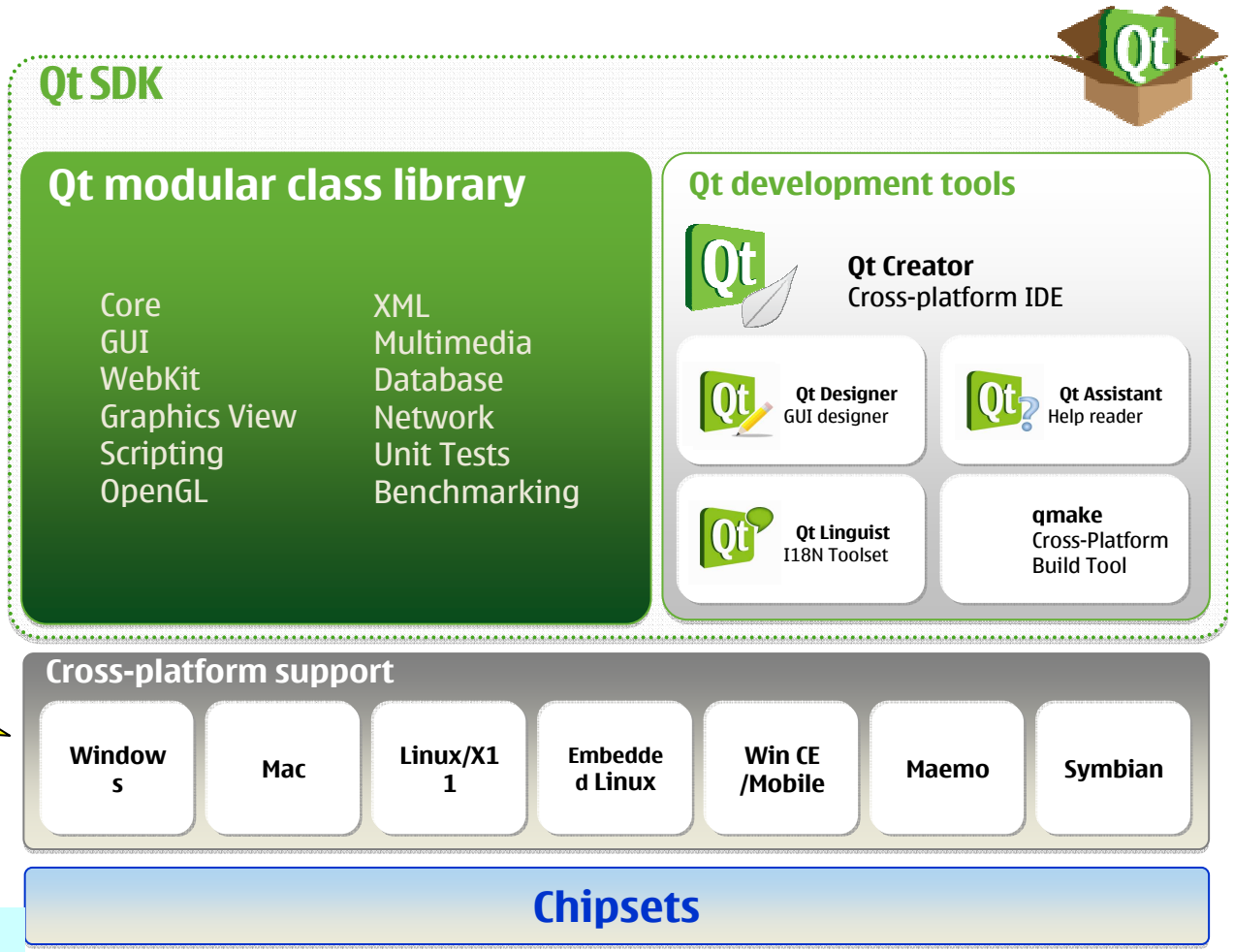
NOKIA

# Simplified offering | Based on Qt

The future offering will be based on Qt, the development tool globally used to build desktop apps

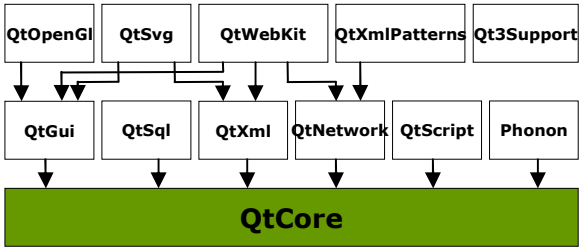
With the simplified toolchain you can tackle f.ex Maemo and Symbian based devices

**Qt 4.6.2**

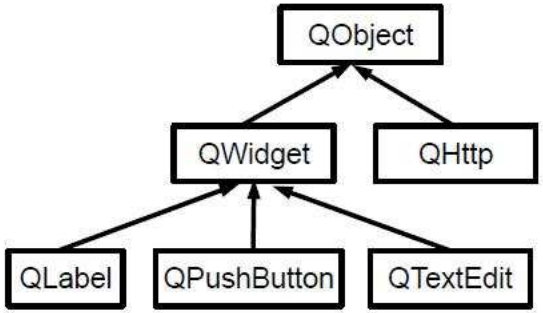



qt.nokia.com/whatsnew

# Qt Core Module

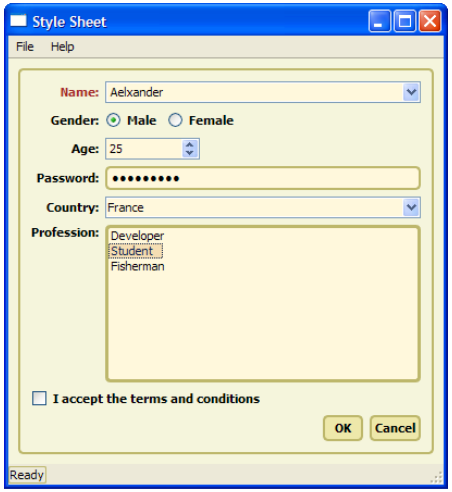


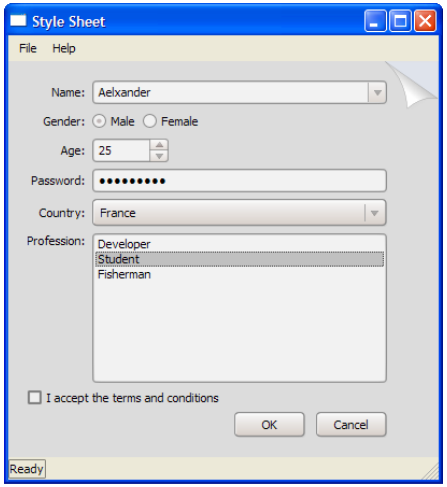

- Classes and methods for basic non-GUI functionality
  - Basic Type (Char, Date, Time, String)
  - File system access, Date and time handling
  - String handling, List and array handling
  - Threads and processes, Shared resources
  - Libraries and Plugins, sophisticated interval driven Timers
- Provides object communication **using signals and slots**
- Queryable and designable **object properties**
  - Q_PROPERTY( bool **focus** READ **hasFocus** )
  - behaves like a class data member
- Contextual **string translation for internationalization**
- Heart of the Qt object model, **QObject**
- Does not depend on underlying window system

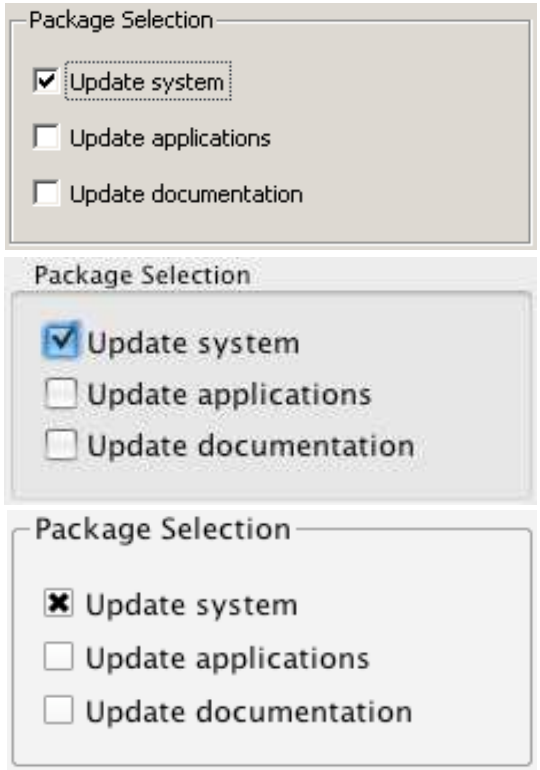


# Qt GUI Classes

- Qt provides rich set of GUI components
  - Simple to complex widgets and controls, dialogs
- Customized components can be written
- Model-View-Controller (MVC) Item views
  - Listviews, treeviews
- Styles – ensuring native or custom look and feel on target platforms
- Font-aware layout system



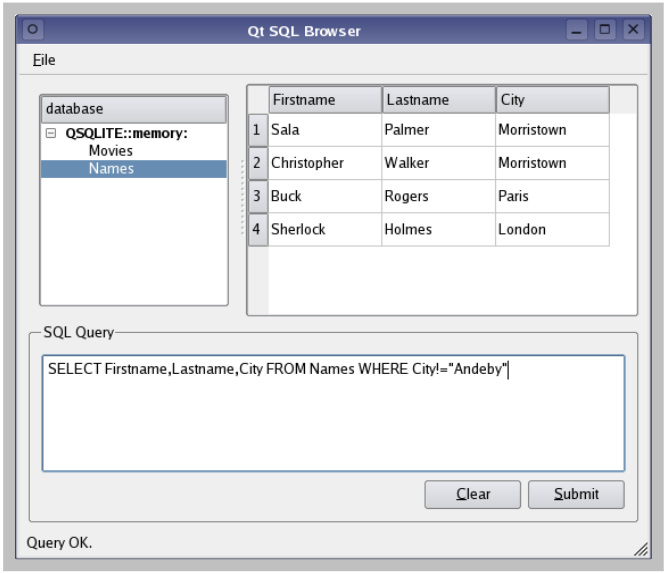




NOKIA

# Qt Database Classes

- Provide platform and database-independent access functionality
- Driver Layer
  - Low-level bridge between specific databases and the SQL API layer
- SQL API Layer
  - Provide access to databases
- User Interface Layer
  - Link data from a database to data-aware widgets
- Supports most major database drivers
  - DB2, IBASE, MySQL, OCI, ODBC, PSQL, SQLITE, TDS




```
db);
```

NOKIA

# Qt Networking Classes

- Provides TCP/IP networking functionality
- TCP sockets for clients and for servers
- HTTP 1.1 compliant asynchronous API
- FTP, DNS implementation & SSL support

```
QHttp http;
connect( http, SIGNAL(done(bool)), this, SLOT(done(bool)) );
http->get( http://www.softafoorumi.com/amazingApi.jsp?action=doGreatThings );

void MyClass::done( bool error )
{
QString data = http->readAll();
 //handle data
}
```

NOKIA

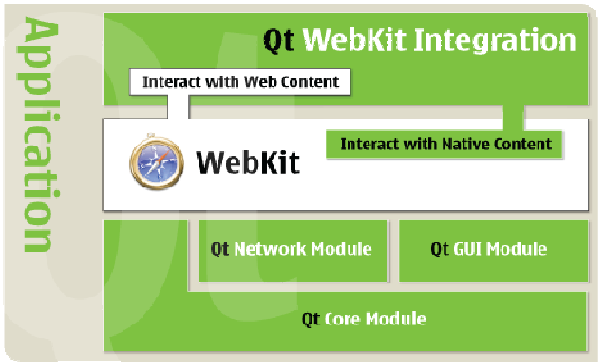
# Qt XML Classes

- Core Module
  - Simple XML stream reader and writer
- XML Module
  - A well-formed XML parser using the SAX2 (Simple API for XML) interface
  - Implementation of the DOM Level 2 (Document Object Model)
- XmlPatterns module
  - An implementation of the XQuery standard
  - Enable users to query XML files similar to SQL
  - Semantics for value assignment, filtering, and simple operations
  - Fully controllable output formatting
- XSLT support

```
QXmlStreamReader xml;
xml.addData( data );
while (!xml.atEnd())
{
 xml.readNext(); ...
// do processing
}
```

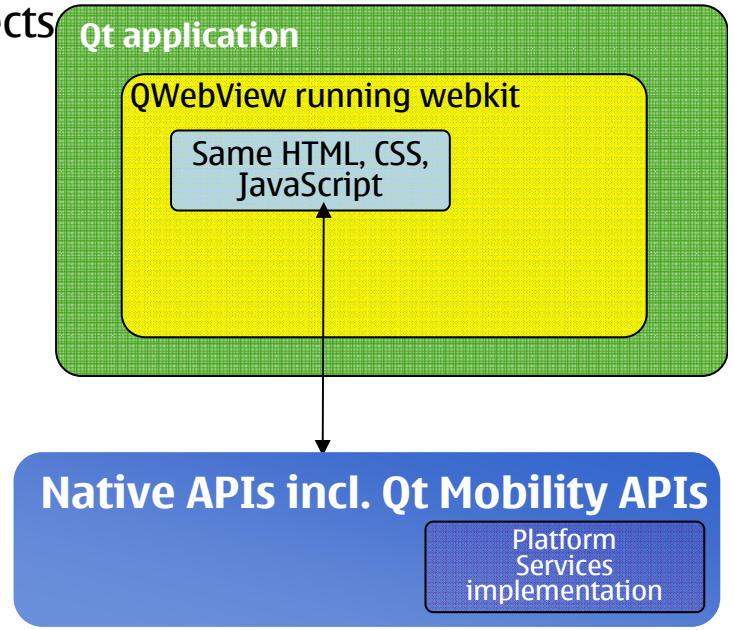
NOKIA

# Qt WebKit Integration




- An open source HTML rendering component integrated with Qt
- Based on webkit (http://webkit.org) that is also used by f.ex Safari, Google Chrome & iPhone
- Web standards compliant
  - Support for HTML, XHTML, XML, stylesheets, JavaScript, HTML editing, HTML canvas, AJAX, XSLT, XPath, some SVG.
- Interact with Web environment, expose native objects

```
QWebView *view = new QWebView( parent );
view->load( QUrl( "http://qt.nokia.com/") );
view->show();
```




qt.nokia.com/forms/whitepapers/reg-whitepaper-webkit-tech

# Do you like money?

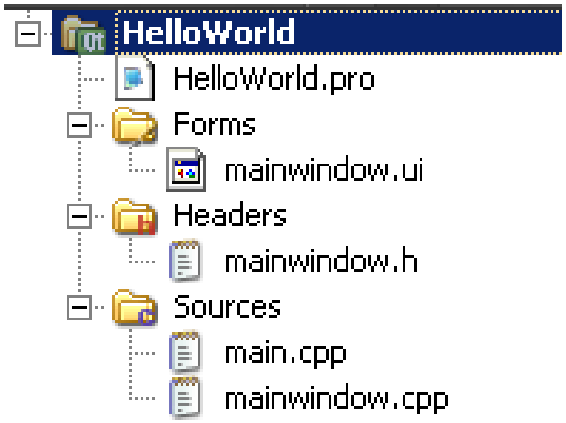

What are you waiting for when you are now facing more than

**+ 130 000 000 Qt-runnable smartphones!**

in addition to all major desktop OSs?

# Basic Project Structure

- Project file
- Headers
- Sources
- Resources
- UI design files

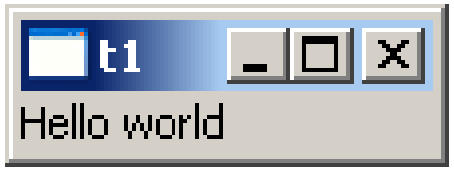



NOKIA

# First Qt Program

You may omit "QtGui" module name if "gui" is included in the .pro file

```
#include <QtGui/QApplication>
#include <QtGui/QLabel>

int main( int argc, char* argv[] )
{
      QApplication hwApp( argc, argv );
      QLabel hwLabel( "Hello world“ );

      // Widgets are not visible by default.
      // Alternatives: showMaximized(), showFullScreen()
      hwLabel.show();
      return hwApp.exec();
}
```




NOKIA

# example of .pro file

```
CONFIG  += qt debug
QT   += network webkit

HEADERS  += hello.h
SOURCES  += hello.cpp
SOURCES  += main.cpp

FORMS  +=
RESOURCES += resource.qrc
TRANSLATIONS  = hello_en.ts \
  hello_zh_CN.ts

win32 {
    SOURCES += hellowin.cpp
}
unix {
    SOURCES += hellounix.cpp
}
!exists( main.cpp ) {
  error( "No main.cpp file found" )
}
win32:debug {
    CONFIG += console
}
```

Specifies for **qmake** that:
- application is based on Qt and
- debug information is needed

Platform specific conditions. This mechanism will be used to specify Symbian/S60 specific details, like required capabilities, UIDs etc.

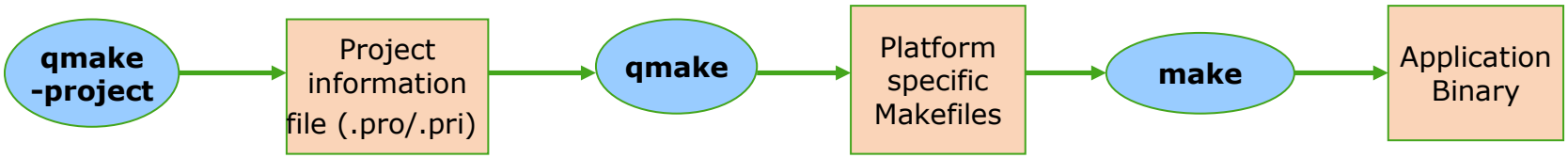
- You may also use flags in your code
- Symbian features or changes flagged with Q_OS_SYMBIAN
- S60 dependencies with Q_WS_S60
- Maemo 5 with Q_WS_MAEMO_5

\Qt\4.6.1\src\corelib\global\qglobal.h

http://doc.qt.nokia.com/4.6/qmake-tutorial.html

# Building Qt Applications

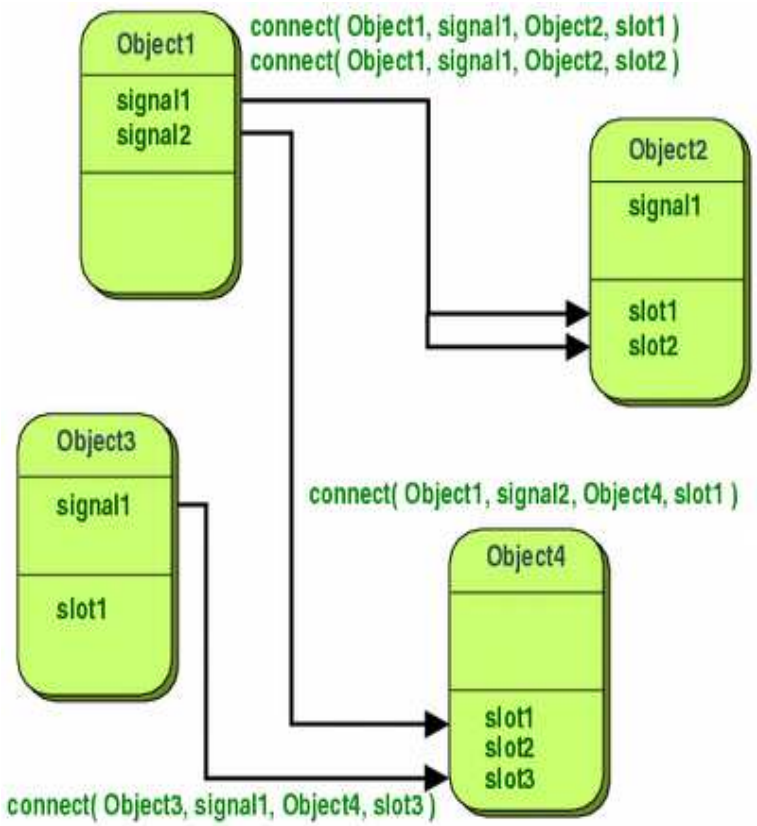
1. **qmake** –project
   - Creates a **Qt project file (.pro)**. This can also be created manually.
2. **qmake**
   - Uses the **.pro** file as input and produces platform-specific Makefile(s) (for Symbian, `bld.inf, .mmp`)
   - Generates make rules to invoke moc for project header files containing ***Q_OBJECT***
3. **make**
   - On Symbian ***`make debug-winscw`*** or ***`make release-gcce`***
   - Compiles the program for the current platform
   - Executes also **moc**, **uic**, **rcc,** or **mifconv**




NOKIA

# Signals and Slots in Core Module

- Inter-object communication mechanism
  - Callback between objects
  - Sender and receiver does not “know about” each other
  - 1-to-many, many-to-1 communication between objects
  - Works across threads
- A signal is emitted when a particular event occurs
- A slot is a function that is called in response to a particular signal.
- A slot may have fewer input parameters than the ones in signal. Extra parameters are ignored




NOKIA

# Signals

- A ***signal*** is a way to inform a possible observer that something of interest has happened inside the observed class
  - A QPushButton is **clicked**
  - An asynchronous service handler is **finished**
  - Value of QSlider is **changed**
- Signals are member functions that are *automatically implemented in the meta-object*
  - Only the function declaration is provided by the developer
- Signal is sent, or *emitted*, using the keyword emit
  - `emit clicked();`
  - `emit someSignal(7, “Hello”); //passing data in a connection`

Signal and slot: http://blog.csdn.net/oowgsoo/archive/2007/03/14/1529284.aspx

NOKIA

# Slots

- A ***slot*** is a function that is to be executed when a signal has been ***emitted***
  - (When QPushButton is pressed), **close** QDialog
  - (When service is ready), **ask for the value and store it**
  - (When QSLider value is changed), **show a new value** in QLCDNumber
- A **slot** function is a normal member function implemented by the developer
- A **slot** could be **private**, **protected** or **public**, and it can be used as a normal function call. Normally, it is of type *void*.

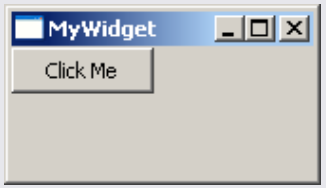
NOKIA

## MyWidget.h

```
class MyWidget : pu
{
    Q_OBJECT

public:
    MyWidget( QWidget *par
);
    ~MyWidget();

signals:
     void quit();

private slots:
    void buttonClicked();

private:
    QPushButton *button;
};
```

- **Important** to define it to have signals/slots mechanism.
- Better to have an independent **header** file instead of having it in the **.cpp** file

```
lget *parent )

Button( "Click Me",  this );

SIGNAL(clicked()), this, SLOT(

}

void MyWidget::buttonClicked()
{
      qDebug( "clicked“ );

            emit quit();
}

MyWidget::~MyWidget()
{
}
```




```
// main.cpp
int main( int argc, char *argv[ ] )
{
     QApplication a( argc, argv
     MyWidget w;
     w.show();
     return a.exec();
}
```

Signal and slot Auto-connection

```
QMetaObject::connectSlotsByName(this);

void on_<object name>_<signal name>(<signal parameters>);
void on_button1_clicked();
```

NOKIA

# UI - QWidget

- Base class for all user interface objects
- A widget without a parent widget is always an independent window
- Example of widgets: QCheckBox, QDateEdit, QLabel, QPushButton, QWebView, QListView, QListView

```
#include <QtGui/QApplication>
#include <QWidget>

int main( int argc, char
*argv[ ]  )
{
    QApplication a( argc, argv
);
    QWidget widget;
    widget.show();
    return a.exec();
}
```

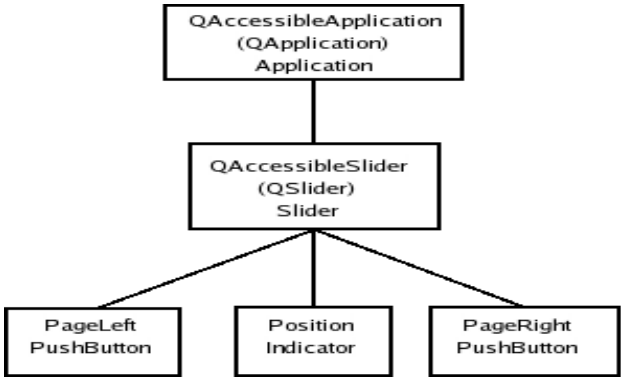

NOKIA

# Parent/Child Relationship

- Each **QObject** instance may take a **parent** (argument for constructor)
- Child informs its parent about its existence, upon which the parent adds it to its own **list of children**
- If a widget object does not have a parent, it is **a window**
- The parent does the following for its children:
  - Deletes them, when it is self-deleted
    - But, of course, is not able to set your own direct pointers to those children to NULL (if not using guarded pointers!)
  - **Hides and shows** children, when hidden/shown itself
  - **Enables and disables** children when enabled or disabled itself
- Note that a child may be explicitly hidden, although the parent is shown

NOKIA

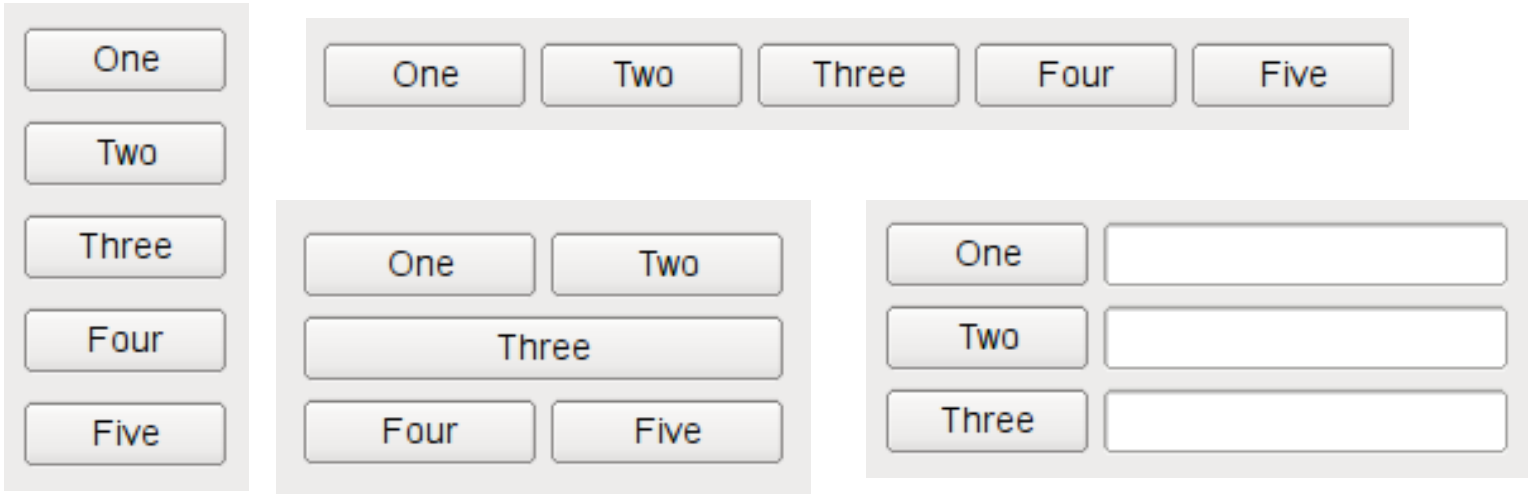
# Memory Management




- The ownership of all child QObjects is transferred to the parent
  - Automatic deletion by the parent
  - Allocated from the heap with ***new***
  - Manual deletion won't however cause double deletion because the child informs its parent of the deletion
- All QObjects without a parent must be deleted manually
  - Stack allocation is a good option to avoid problems
- Occasionally it may seem like Qt would hold some sort of automatic garbage collection but this is not true!
  - Always pay attention to ownerships and responsibilities!

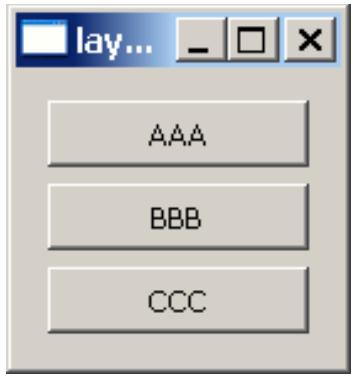
# UI - Layouts

- Layout classes
  - Easy way of arranging child widgets within widget
  - Available layouts: QHBoxLayout, QVBoxLayout, QGridLayout, QFormLayout, QStackedLayout
  - You may create your own layout if necessary such as flow layout




NOKIA

# UI - Layouts




```
int main( int argc, char *argv[ ] )
{
    QApplication app (argc, argv );    //1

    QWidget window;   //2
    QPushButton *buttonA = new QPushButton( "AAA“ );
   buttonA->setObjectName("AAA");
    QPushButton *buttonB = new QPushButton( "BBB“ );
    buttonB->setObjectName("BBB");
    QPushButton *buttonC = new QPushButton( "CCC“ );
   buttonC->setObjectName("CCC");

    QVBoxLayout *layout = new QVBoxLayout;  //4
   layout->setObjectName("layout");
    layout->addWidget( buttonA );  //5
    layout->addWidget( buttonB );
    layout->addWidget( buttonC );
    layout->addWidget( buttonD );

    window.setLayout( layout );  //6

    window.dumpObjectTree();
    app.dumpObjectTree();

    window.show();

    return a.exec();   //7
}
```

**“window” Object tree:**

```
QWidget::
  QWidget::
  QVBoxLayout::layout
  QPushButton::AAA
  QPushButton::BBB
  QPushButton::CCC
```

**“app” object tree:**

```
QApplication::layouttest
   QEventDispatcherWin32::
   QWindowsXPStyle::windowsxp
   QSessionManager::qt_sessionmanager
```

NOKIA

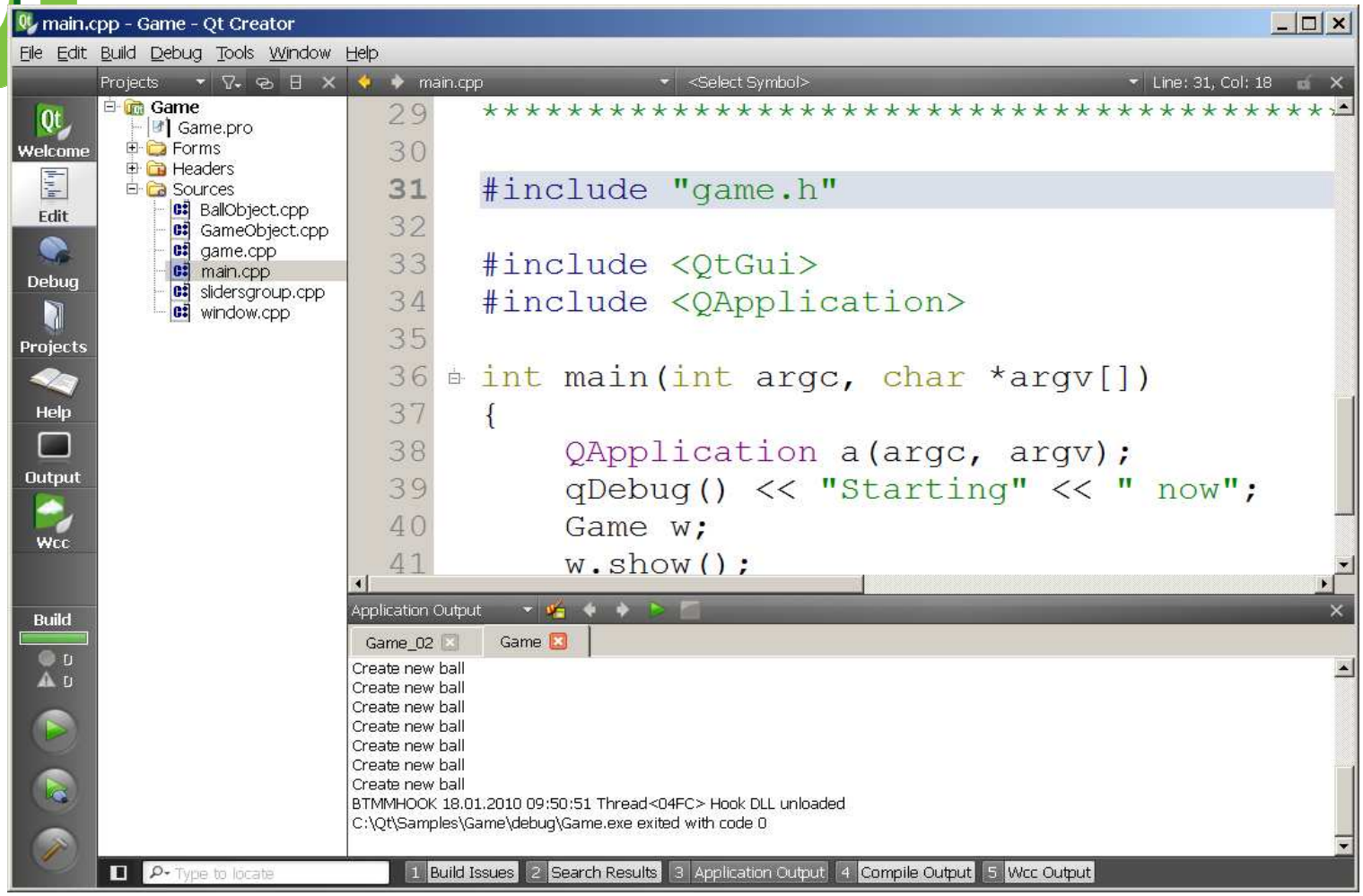
# Qt dev environment setup for Symbian

- Set up environment for Symbian/S60 development
- Set up environment for Qt on PC
  - qt.nokia.com
- Installations for the target device

- Qt for Symbian Developer's Library
- http://labs.trolltech.com/blogs/ (Videos)
- http://wiki.forum.nokia.com/index.php/Qt_for_Symbian
- http://wiki.forum.nokia.com/index.php/Installing_Qt_on_Symbian
- http://wiki.forum.nokia.com/index.php/Qt_Tutorial_Lesson_1:_Installation
- Setting up for the Qt for Symbian development environment

NOKIA

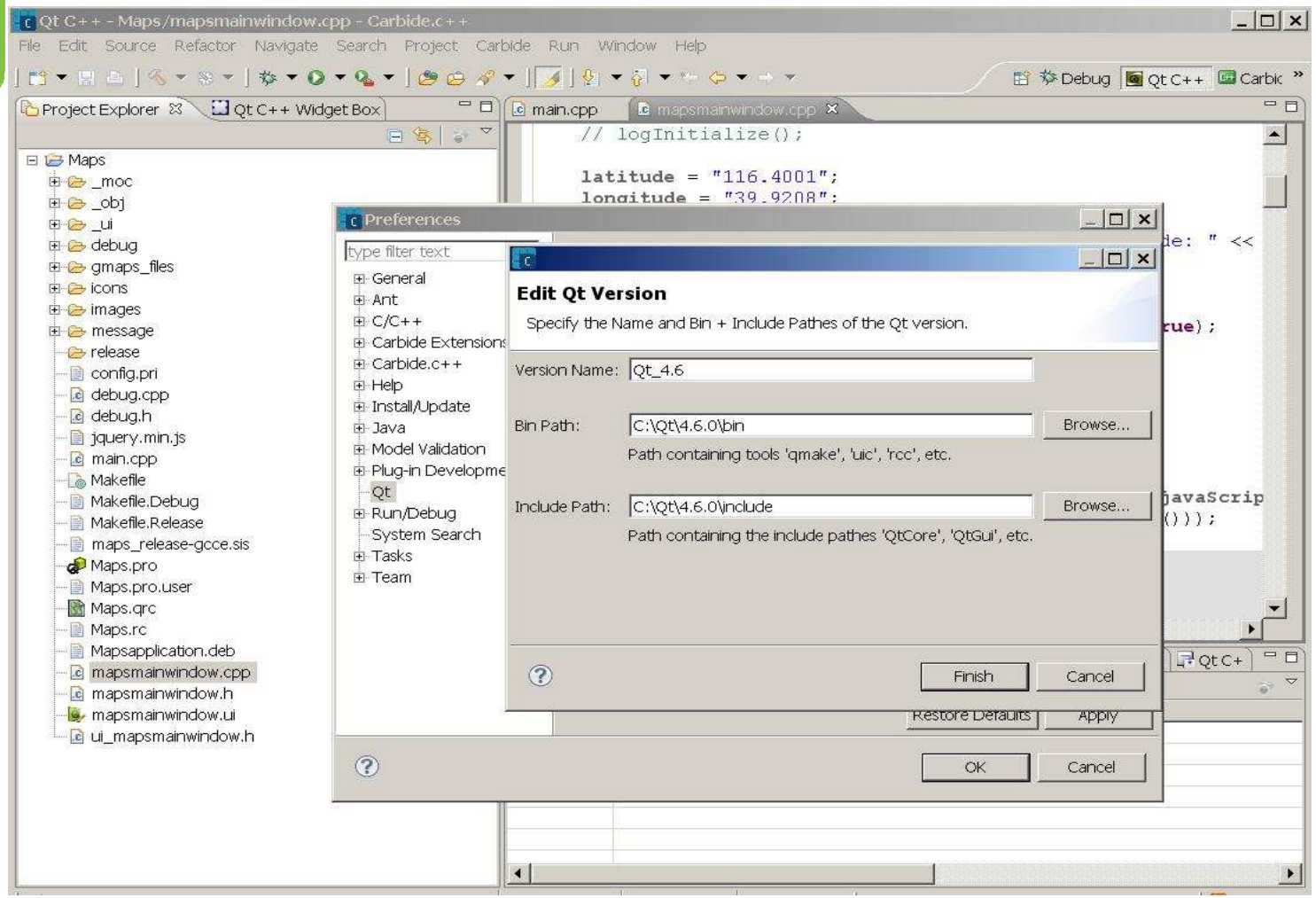
# Development tools – Qt creator




http://wiki.forum.nokia.com/index.php/How_to_use_Qt_Creator_IDE

NOKIA

# Development tools – Carbide.c++ 2.3/2.4




http://wiki.forum.nokia.com/index.php/Getting_started_with_Qt_for_Symbian

NOKIA

# make command

- Running **make** command calls **bldmake** and **abld** to build the Qt application for the S60 environment

```
Add lines to C:\Qt\4.6.2\bin\qtenv.bat

set QT_SIS_CERTIFICATE=C:\Qt\mycer.cer
set QT_SIS_KEY=C:\Qt\mykey.key
set QT_SIS_PASSPHRASE=12345

Change the path for your certificate
```

```
createpackage -i contactsex_gcce_urel.pkg
or
make sis
```

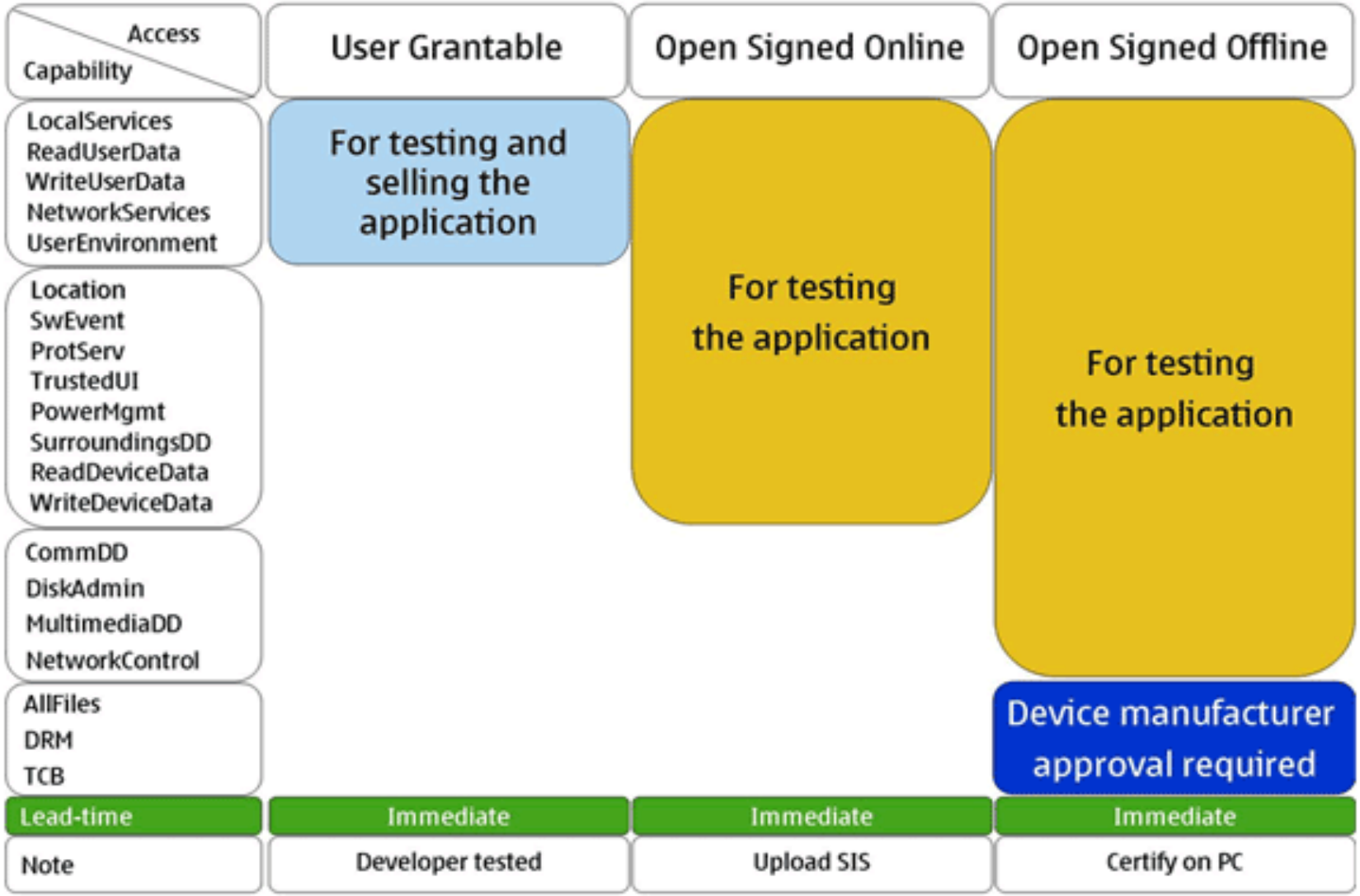
| Command | Description |
|---|---|
| make | Creates abld and the project makefiles and builds debug build of the application for the emulator (winscw udeb). |
| make debug | Creates debug builds (winscw/gcce/armv5 udeb). |
| make debug-winscw | Creates winscw debug build. |
| make debug-gcce | Creates gcce debug build. |
| make debug-armv5 | Creates armv5 debug build. |
| make release | Creates release builds (gcce/armv5 urel). |
| make release-gcce | Creates gcce release build. |
| make release-armv5 | Creates armv5 release build. |
| make export | Copies the exported files to their destination. |
| make cleanexport | Removes files created with `make export`. |
| make mocclean | Removes the header and source files created by the moc tool. |
| make mocables | Runs moc tool on necessary files. |
| make clean | Removes everything built with abld target, exported files and makefiles. |
| make distclean | As make clean, but also removes all Symbian specific files created with qmake. |
| make confclean | As make distclean, but also cleans everything generated by configure call. Note that this command is only available in the Qt root directory. |
| make run | Launches the application in the emulator. |

NOKIA

# Symbian Platform Security

- All platform security rules also apply for Qt applications in the Symbian environment.
- Qt is mainly ported on top of Open C/C++, the required capabilities are also derived from those APIs. Platform security requires that needed capabilities be defined in the project file. The Qt application may require, for example, the following capabilities:
  - **AllFiles**, when using file operations and accessing protected folders
  - **NetworkServices** should be enough in most cases when using the **QtNetwork** module, but there might be certain API calls that also require **NetworkControl**
- When using Symbian APIs the capabilities needed are, of course, the ones that the APIs define

NOKIA


Access
Capability
User Grantable
Open Signed Online
Open Signed Offline
LocalServices
ReadUserData
WriteUserData
NetworkServices
UserEnvironment
For testing and selling the application
Location
SwEvent
ProtServ
TrustedUI
PowerMgmt
SurroundingsDD
ReadDeviceData
WriteDeviceData
For testing the application
CommDD
DiskAdmin
MultimediaDD
NetworkControl
For testing the application
AllFiles
DRM
TCB
Device manufacturer approval required
Lead-time
Immediate
Immediate
Immediate
Note
Developer tested
Upload SIS
Certify on PC

NOKIA

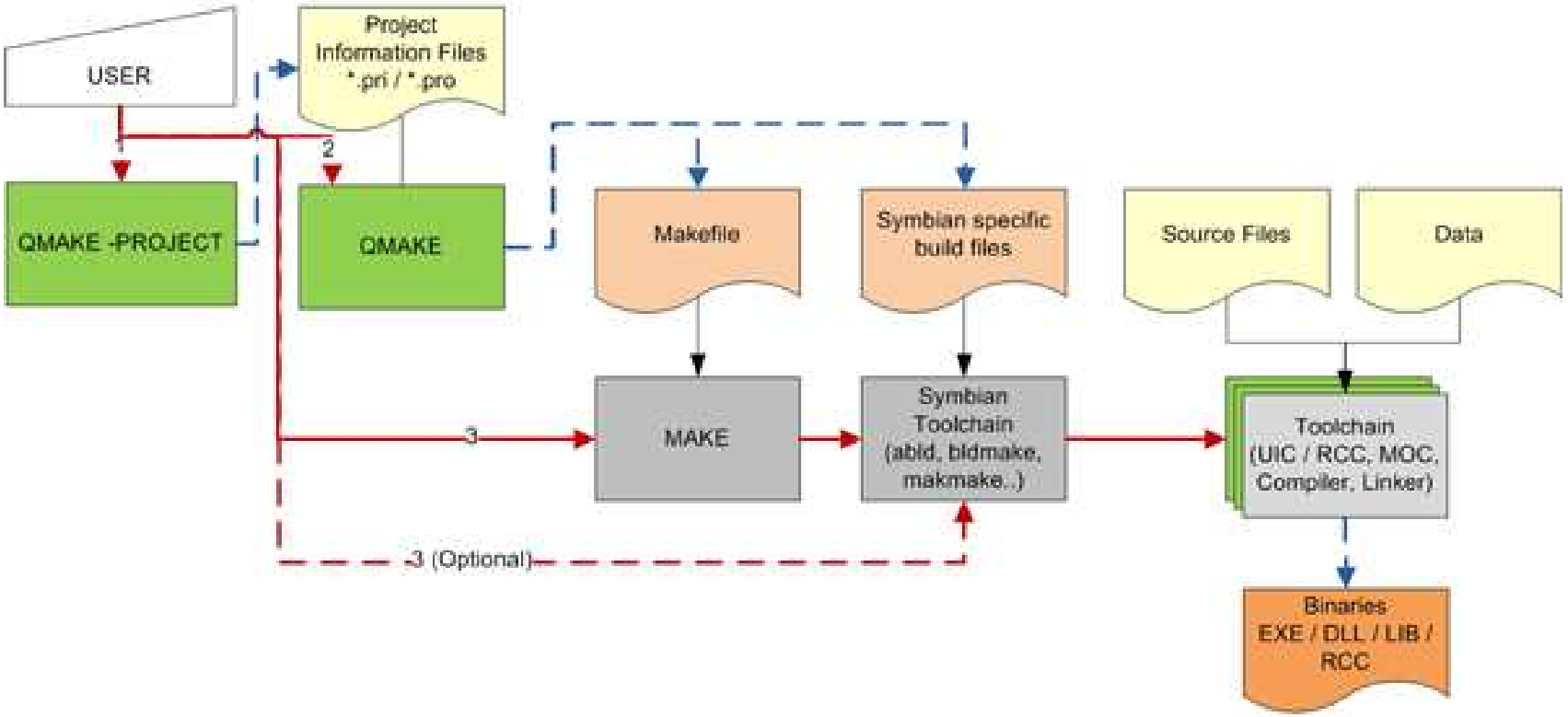
# Symbian Platform specific

- Capabilities are not defined in MMP file but in PRO file

```
TARGET = SimpleWidget                          ← target name
TEMPLATE = app                                 ← target type

HEADERS  += mainwindow.h                       ← header files

SOURCES += main.cpp\                           ← source files
      mainwindow.cpp

Symbian {                                      ← Symbian specific
  LIBS += -llbs  \                             ← libraries for Symbian
          -letel3rdparty
  TARGET.SID = 0xA000017F                      ← Security ID
  TARGET.VID = 0x70000001                      ← Vendor ID
  TARGET.CAPABILITY = Location networkservices ← Capabilities
  TARGET.EPOCSTACKSIZE = 0x5000                ← Stack size
  // Min 128Kb, Max 16Mb
  TARGET.EPOCHEAPSIZE = "0x20000 0x1000000"    ← Heap size
  ICON = ./images/myIcon.svg                   ← App icon
}
```


bool QApplication::symbianEventFilter(const QSymbianEvent * event )

# Qt and Symbian OS Tool Chain Integration




NOKIA

# “Smart Installer” in Brief

**Why?**

- Qt libraries are not part of the older S60 devices (S60 3.1-5.0)
- Total size of the Qt libraries is significant (10-12MB)
  ➔ Not feasible to include Qt libraries with every application based on Qt
- Solution: “**Smart Installer**”

**How?**

- Toolchain integrated utility to help developers in Qt based application packaging and distribution
  - A small utility (less than 20 kB) is included automatically within the installation package (.sis) which is then launched first during the installation. This utility is the so called “Smart Installer”
- The utility checks if the required libraries are already installed on the device
  - Required, but missing libraries are fetched over the network during installation, *but only if needed*. User has the option to download or not, however, the installation is cancelled if no permission is given to download the required libraries.
  - Requires network availability during installation (packet data or WLAN)
  - Also upgrades the libraries, if necessary

NOKIA

# ”Smart installer” - Qt installer for Symbian

**The opportunity**
+130M Symbian devices could run cross-platform Qt applications already during this year

**The problem**
Qt libraries need to be post installed – most consumers might not bother

**The solution**
A Smart Installer tool that checks whether needed Qt version is available on the device – and if not, it will handle the installation

## Details

- Nokia is working on a tool called Smart Installer for Symbian devices
- The expected availability is **1Q 2010**
- **A piece of code is to be packaged together with the Qt application targeted for Symbian devices (when creating the .sis file)**
- The Smart Installer will download OTA the needed Qt version to the handset if it is not already present

http://qt.nokia.com/developer/nokia-smart-installer-for-symbian

# Sample application.pkg file

UID of real app package

Qt verion dependency

```
&EN
; SIS header: name, uid, version
#{"CallEvent"},(0xECBC4088),1,0,0

[0x101F7961],0,0,0,{"S60ProductID"}

; Localised Vendor name
%{"Forum Nokia, XiaoGuo"}
; Unique Vendor name
:"Forum Nokia, Beijing, China"

; Default dependency to Qt libraries
(0x2001E61C), 4, 6, 2, {"Qt"}

; Executable and default resource files
"/Symbian/S60_5th_Edition_SDK_v1.0/epoc32/release/gcce/urel/CallEvent.exe"    -
  "!:\sys\bin\CallEvent.exe"
"/Symbian/S60_5th_Edition_SDK_v1.0/epoc32/data/z/resource/apps/CallEvent.rsc"    -
  "!:\resource\apps\CallEvent.rsc"
"/Symbian/S60_5th_Edition_SDK_v1.0/epoc32/data/z/private/10003a3f/import/apps/CallEv
  ent_reg.rsc"- "!:\private\10003a3f\import\apps\CallEvent_reg.rsc"
"/Symbian/S60_5th_Edition_SDK_v1.0/epoc32/data/z/resource/apps/CallEvent.mif"    -
  "!:\resource\apps\CallEvent.mif"
```

NOKIA

# Example of a installer package file

```
; Language
&EN

; SIS header: name, uid, version
#{"CallEvent installer"},(0xA000D7CE),1,0,0

[0x101F7961],0,0,0,{"S60ProductID"}
[0x102032BE],0,0,0,{"S60ProductID"}
[0x102752AE],0,0,0,{"S60ProductID"}
[0x1028315F],0,0,0,{"S60ProductID"}

; Localised Vendor name
%{"Forum Nokia, XiaoGuo"}
 ; Unique Vendor name
:"Forum Nokia, Beijing, China"

; Default dependency to Qt libraries
"C:/CodeCamp/Qt/demos/CallEvent_v2/CallEvent.sis"
  "c:\adm\CallEvent.sis"
@"C:/Qt/4.6.2/smartinstaller.sis",(0x2002CCCD)
```

**UID of installer package**

Actual application sis file

UID of smartinstaller.sis

# The Mobility Project (1/3)

- What is it?
  - New Qt APIs enabling cross-platform mobile application development and service access.

- What is the value?
  - Significant advantage for developers targeting mobile platforms, such as Windows CE, Symbian, and Maemo in 2010.
    - Significant advantage for carriers and their 3rd party content developers
    - Easier to create to applications targeting many platforms
  - Reuse of code between the mobile platforms
  - Bring existing Qt developers and new ideas to Nokia platforms.
  - Bring Nokia developers to non-Nokia platforms.

- http://wiki.forum.nokia.com/index.php/Qt_Mobility_Technology_Preview
- http://wiki.forum.nokia.com/index.php/Mobile_Extensions
- http://wiki.forum.nokia.com/index.php/Qt_Mobility_Contest

NOKIA

# The Mobility Project (2/3)

- Service Framework
  - Launch, discover, and communicate with services
  - Use services natively or through a run-time language such as Javascript
  - Control access to services
- Context Framework/Publish and Subscribe
  - Share context information between applications
- Contacts API
  - Access stored contacts
  - Create new contacts
- Location API
  - Query current location
  - API hides underlying Location source (GPS, Cell ID, etc)

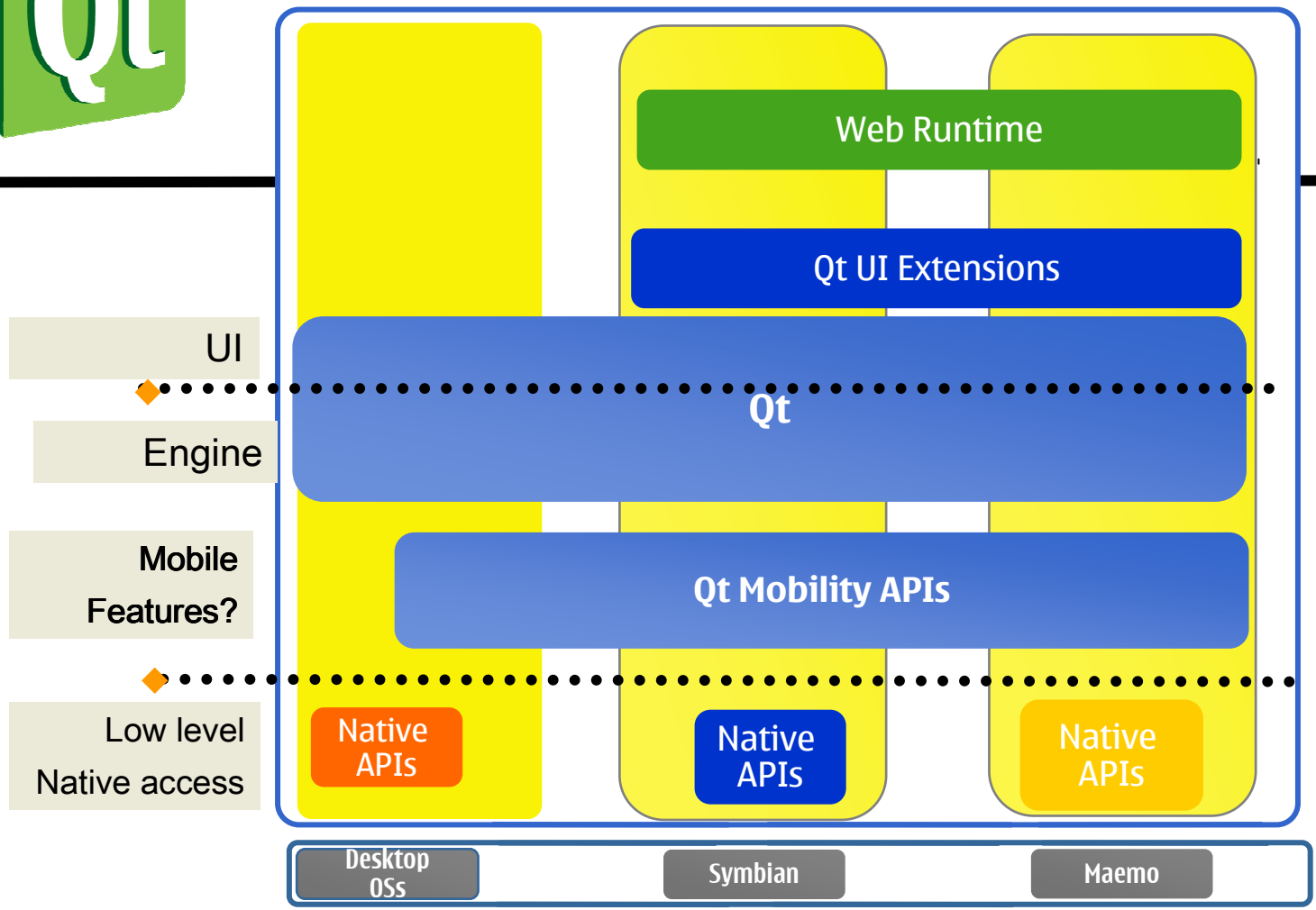
NOKIA

# The Mobility Project (3/3)

- System Information API
  - Access to platform and/or device information
  - Determine status of available resources
- Bearer Management
  - Manage available network connections
  - Use the best available connection
- Messaging API
  - Send and receive a variety of message types
- Multi-Media API
  - Play and record audio/video

NOKIA

# Qt and Web are the ways to go




NOKIA

# Roadmap

## Public

| | Delivered Features | 2H'2009 | 2010+ |
|---|---|---|---|
| Framework | **Qt Framework 4.5**<br>• Improved graphics painting performance<br>• Mac OS X Cocoa Framework<br>• WinCE Phonon + webkit<br>• Webkit: NPAPI, client side storage, Javascript SQL, Multimedia elements | **Qt Framework 4.6**<br>• Animation API<br>• States & Transitions<br>• Multi-touch & Gestures<br>• OpenVG<br>• JavaScript Unification<br>• jQuery inspired DOM access<br>• 3D enablers<br>• S60 as a new platform<br>• Windows 7 and Mac OS X 10.6 support | **Committed features**<br>• Declarative UI<br>**Framework Research**<br>• Media services<br>• Hybrid application development<br>• Memory and resource handling<br>• Qt/3D portability API<br>• Next generation item views<br>• New Qt APIs for mobile development |
| Tools | **Qt Tools**<br>• Qt Creator 1.2 (Qt IDE) | **Qt Tools**<br>• Qt Creator 1.3: Basic support for Symbian development<br>• Continue and enhance the VS add-in and Eclipse plug-ins | **Committed features**<br>• Declarative UI designer<br>**Tools Research**<br>• Qt Creator 2.0<br>• Hybrid application development<br>• Build systems<br>• Community and collaboration tools |

NOKIA

# Qt 4.7

- Declarative UI
- No new platforms
  - Better support for Windows Mobile 6.5
- http://blog.qt.nokia.com/2009/10/21/qt-4-7-is-in-the-works/
- Will be integrated to Symbian^4

NOKIA

# Demos

**N97:**
Maps
Qsimpleimageviewer
MobilePaint
Qt-labs demos

**N900**
Collidingmice
Centralplace
Maps
Qt-labs demos

**Windows**
Maps
Collidingmice


**Qt4.6 Mobile Demos on Symbian/ Maemo:**

http://www.youtube.com/watch?v=PCx8RfNhhXk

http://wiki.forum.nokia.com/index.php/Qt_sample_application:_Google_maps