Archived:Setting up environment for Qt Mobility API

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*}}.

The Qt Mobility APIs are now present in the Qt SDK, which is the only SDK available for mobile development.

This article explains how to set up a Symbian SDK to work with Qt Mobility APIs (this is no longer necessary because the APIs are included in the Qt SDK). Deprecated)

Prerequisites

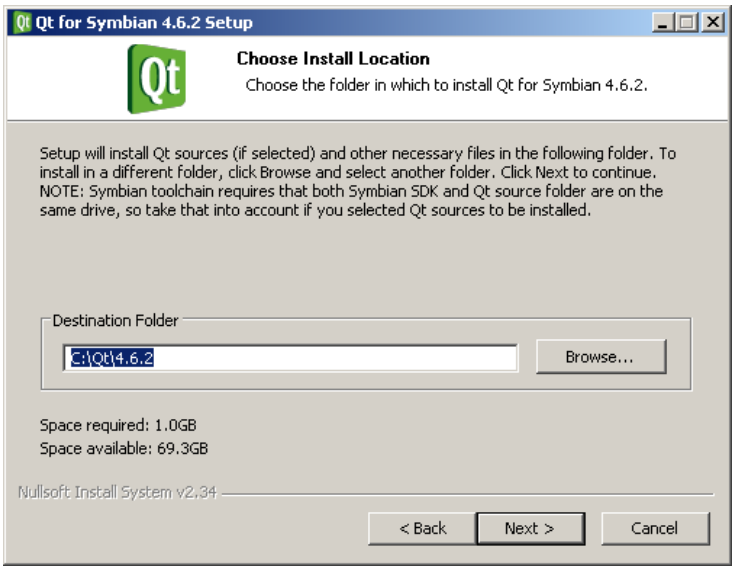
- Install the S60 SDK: S60 Platform and Device SDKs for Symbian OS. (In this article, S60 5th Edition SDK is used)
- Use Carbide.c++ or Qt Creator as an IDE ((In this article, Carbide.c++ v2.4 is used).
  - To set up Carbide.c++, first install Carbide.c++ v2.3.
  - Upgrade it to latest version as shown in this blog post.

Steps

Setup Qt

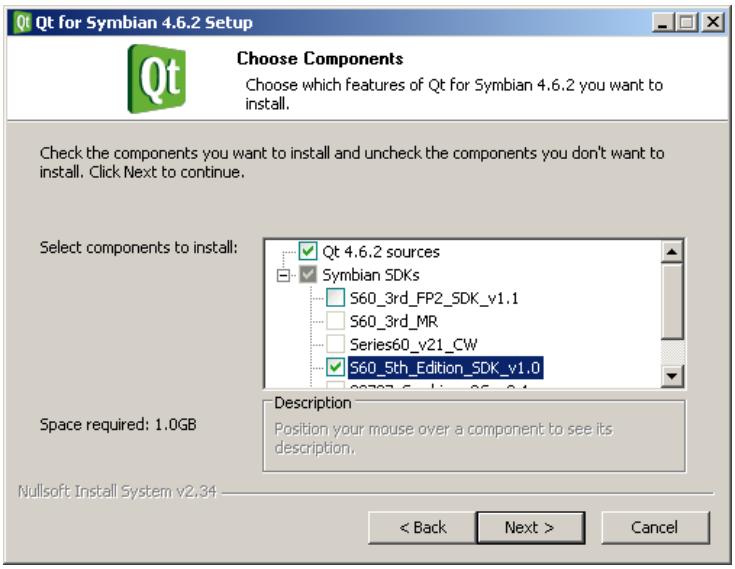
- Download Qt.
- Install it on your PC.

- Choose the path of installation

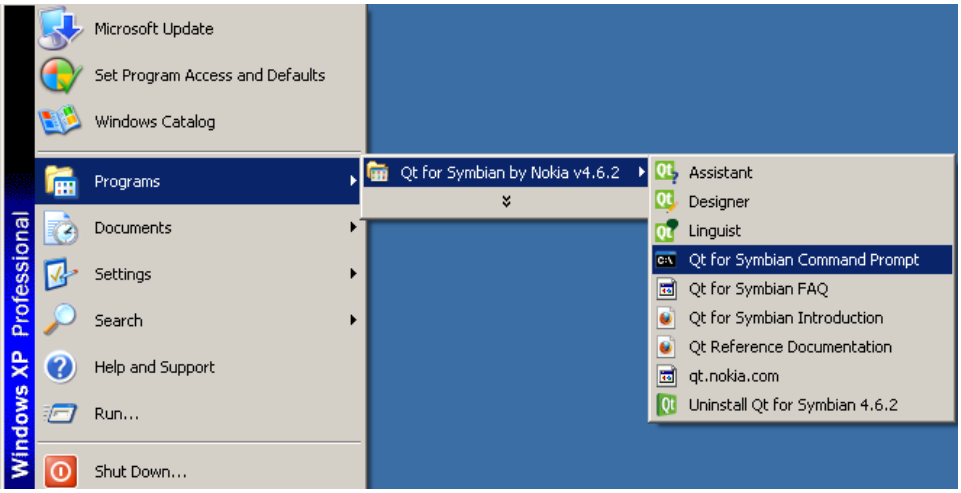


(C) Copyright Nokia 2013. All rights reserved.

- Choose the Symbian SDK(s)



- Qt provides a command prompt configured with the correct paths and environment variables for building Qt applications from the command line:



Note: Before testing a Qt application on a (S60 3rd Edition or 5th Edition) phone, Qt libraries must be installed on the device. Easiest way to do this is to connect the device to the PC using Nokia PC Suite and use the ***qt_installer.sis***, located in the root folder of Qt installation. This .sis file will install all Qt libraries and dependencies to an S60 device.

Setup Qt Mobility API

- After successful installation of Qt, the next step is to set up Qt Mobility
- Download QtMobility API (also known as *New Qt APIs*) source and binary package for Symbian.
- **Edited by Rahul** - I downloaded from here QTMobilitySymbian API above link was not having zip files required in next steps.
- Extract the file to a suitable location. Here, `C:\Qt` is used.
- Inside the package, you'll find the following files:
- - `qt-mobility-1.0.0-epoc32-3.1.zip`
  - `qt-mobility-1.0.0-epoc32-3.2.zip`
  - `qt-mobility-1.0.0-epoc32-5.0.zip`
  - `convenienceheaders.zip`
  - `qtmobility.sis`
- Extract `qt-mobility-1.0.0-epoc32-<S60 SDK>.zip` packages into the root folder of the respective S60 SDK.
- - For example, if using S60 5th Edition SDK, extract `qt-mobility-1.0.0-epoc32-5.0.zip` into `C:\S60\devices\S60_5th_Edition_SDK_v1.0`. Repeat this step for all the Symbian SDKs you selected during Qt installation.

- Extract `convenienceheaders.zip` into the root folder of each S60 SDK (as above).
  - This will enable Qt style include statements (e.g. `#include <QSystemInfo>` instead of `#include <qsysteminfo.h>`).

- Install `qtmobility.sis` file to device. You can do this, for example, via bluetooth or using Nokia PC Suite (recommended).

(C) Copyright Nokia 2013. All rights reserved.

- Supported devices: S60 3rd Edition, Feature Pack 1 or later. see http://www.developer.nokia.com/Devices/ for device-specific information

- Copy (and rename) `C:/Qt/qt-mobility-symbian-opensource-1.0.0/features/mobility.prf.template` file to `<QTDIR>/mkspecs/features/mobility.prf`
  - <QTDIR> can be obtained by running 'qmake -v' from the Qt command prompt.

You now have a working environment for building Qt projects that use QtMobility libraries.

Building the examples

To test that you have a working environment, try building the examples from Qt command prompt:

```
C:/Qt/qt-mobility-symbian-opensource-1.0.0> configure -examples
```

If configure completes successfully, build the examples:

```
C:/Qt/qt-mobility-symbian-opensource-1.0.0> cd examples
C:/Qt/qt-mobility-symbian-opensource-1.0.0/examples> qmake
C:/Qt/qt-mobility-symbian-opensource-1.0.0/examples> make release-gcce
```

To generate a sis package and automatically deploy it to a device connected to PC Suite, use `createpackage` command with `-i` option. For example:

```
C:/Qt/qt-mobility-symbian-opensource-1.0.0/examples> cd bearercloud
C:/Qt/qt-mobility-symbian-opensource-1.0.0/examples/bearercloud> createpackage -i bearercloud_template.pkg release-gcce
```

Note: Using `createpackage` without *certificate* option will sign the application with a self-signed certificate. Only those examples that use basic capabilities (`TARGET.CAPABILITY` in .pro file), granted by the user during installation, can be self-signed. See Capabilities for more information.

Related articles

- Getting started with Qt Mobility APIs
- Working with Carbide.c++ IDE for Qt Mobility APIs
- Working with QSystemInfo - System Information API - Part 1
- Working with QSystemInfo - System Information API - Part 2
- Working with QSystemDeviceInfo - System Information API - Part 1
- Working with QSystemStorageInfo - System Information API
- Working with QSystemDisplayInfo - System Information API
- Working with QSystemNetworkInfo - System Information API - Part 1
- Working with QSystemNetworkInfo - System Information API - Part 2

Reference links

- Qt - cross-platform application and UI framework
- Qt Mobility API
- New Qt APIs Beta - Mobility Project
- SDK help

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.