

“Write once, compile **anywhere**”

# Basic Programming Tutorial

Han Sol Kang

# Contents



• Introduction

• Installation

• Qt application

• Example

# Introduction



Qt (/kju:t/ "cute") : a cross-platform application framework that is widely used for developing application software

“Write **once**, compile **anywhere**” (WOCA)

“Q”

Haavard's Emacs typeface

“t”

Inspired by Xt (X toolkit)



Haavard Nord(left) and Eirik Chambe-Eng(right)

# Introduction



## The future is written with Qt

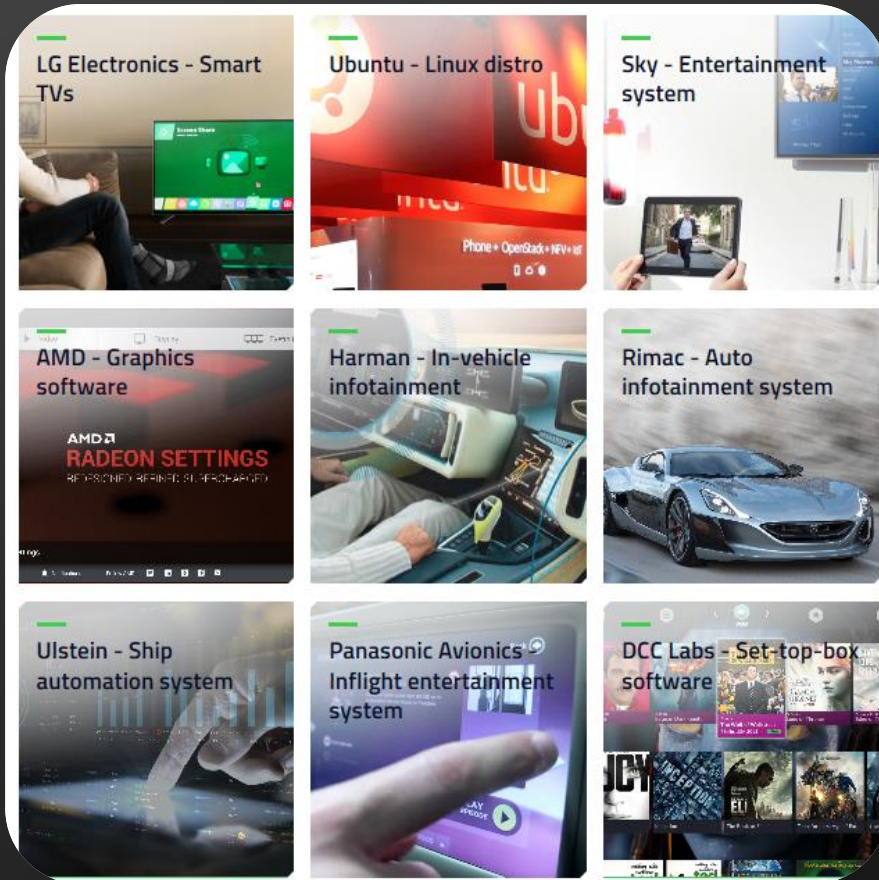
Cross-platform development with Qt is the faster, smarter way to create innovative UIs, applications & embedded devices.

A modern user interface that is beautiful on every screen and performs perfectly on every platform is not an option, it's a necessity.

— 8 of Top 10 Fortune 500 companies use Qt —



# Introduction



If you want to know just what's possible with Qt, all you have to do is look at how some of the leading companies **in more than 70 industries are powering millions of devices** all over the world including Rimac, Formlabs, AMD, Ubuntu, Imaginando, Xsens, Dolby Labs, Tableau, Holoplot, Ulstein, NXP, and many more.

# Introduction

## ❖ Supported Platforms (Officially)

<u>Windows</u>	
Windows 10 (64-bit)	<b>MSVC 2015</b>
Windows 10 (32-bit)	MSVC 2015
Windows 8.1 (64-bit)	MSVC 2015, <b>MSVC 2013</b> , <a href="#">MinGW</a> 5.3, <a href="#">MinGW</a> 4.9, <a href="#">MinGW</a> 4.8
Windows 8.1 (32-bit)	MSVC 2015, MSVC 2013, <a href="#">MinGW</a> 5.3, <a href="#">MinGW</a> 4.9, <a href="#">MinGW</a> 4.8
Windows 7 (64-bit)	MSVC 2015, MSVC 2013, <a href="#">MinGW</a> 5.3, <a href="#">MinGW</a> 4.9, <a href="#">MinGW</a> 4.8
Windows 7 (32-bit)	MSVC 2015, MSVC 2013, <a href="#">MinGW</a> 5.3, <a href="#">MinGW</a> 4.9, <a href="#">MinGW</a> 4.8

<u>Linux/X11</u>	
openSUSE 13.1 (64-bit)	<b>GCC 4.8.1</b>
Red Hat Enterprise Linux 6.6 (64-bit)	<b>GCC 4.9.1</b>
Ubuntu 14.04 (64-bit)	<b>GCC 4.6.3</b>
(Linux 32/64-bit)	GCC 4.8.1, GCC 4.9.1

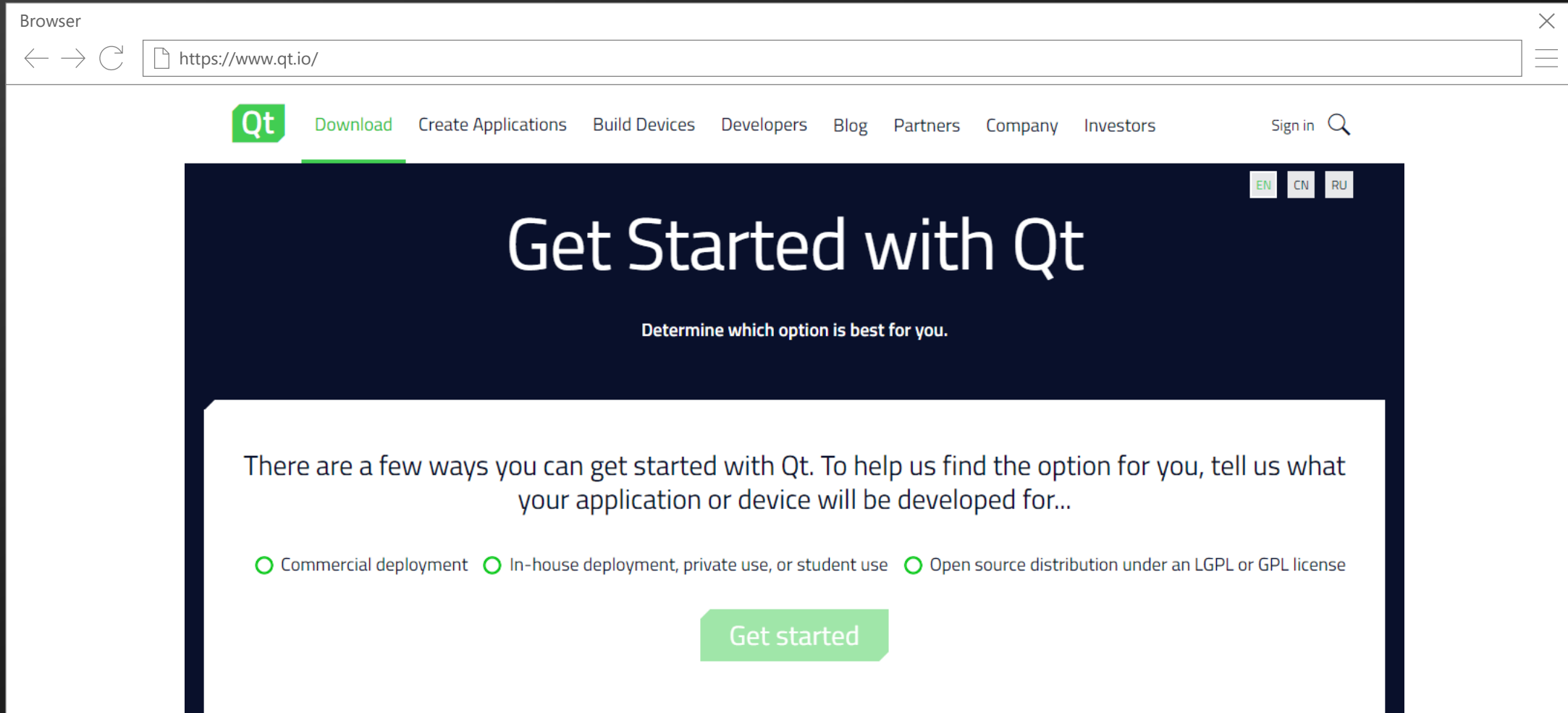
<u>OS X</u>	
OS X 10.8, 10.9, 10.10, 10.11	<b>Clang as provided by Apple</b>

<b>Embedded Platforms: <a href="#">Embedded Linux</a>, <a href="#">QNX</a></b>	
<a href="#">Embedded Linux</a>	GCC
QNX 6.6.0 (armv7le and x86)	<b>As provided by QNX</b>

<b>Mobile Platforms: <a href="#">Android</a>, <a href="#">iOS</a>, <a href="#">WinRT</a></b>	
Windows Phone 8.1 (arm)	<b>MSVC 2013</b>
Windows Runtime (x86, <b>x86_64</b> , arm)	MSVC 2015, <b>MSVC 2013</b>
iOS 6 and above	<b>Clang as provided by Apple</b>
Android (API Level: 16)	<b>GCC as provided by Google</b>

# Installation

## ❖ Download Qt



The screenshot shows a web browser window displaying the Qt website. The address bar shows the URL `https://www.qt.io/`. The navigation menu includes links for [Download](#), [Create Applications](#), [Build Devices](#), [Developers](#), [Blog](#), [Partners](#), [Company](#), and [Investors](#). There is also a [Sign in](#) link and a search icon. The main content area features a dark blue header with the Qt logo and the text "Get Started with Qt". Below this, it says "Determine which option is best for you." and provides three radio button options: "Commercial deployment", "In-house deployment, private use, or student use", and "Open source distribution under an LGPL or GPL license". A green "Get started" button is positioned at the bottom of the form.

Browser

← → ↻ `https://www.qt.io/`

Qt Download Create Applications Build Devices Developers Blog Partners Company Investors Sign in 🔍

EN CN RU

## Get Started with Qt

Determine which option is best for you.

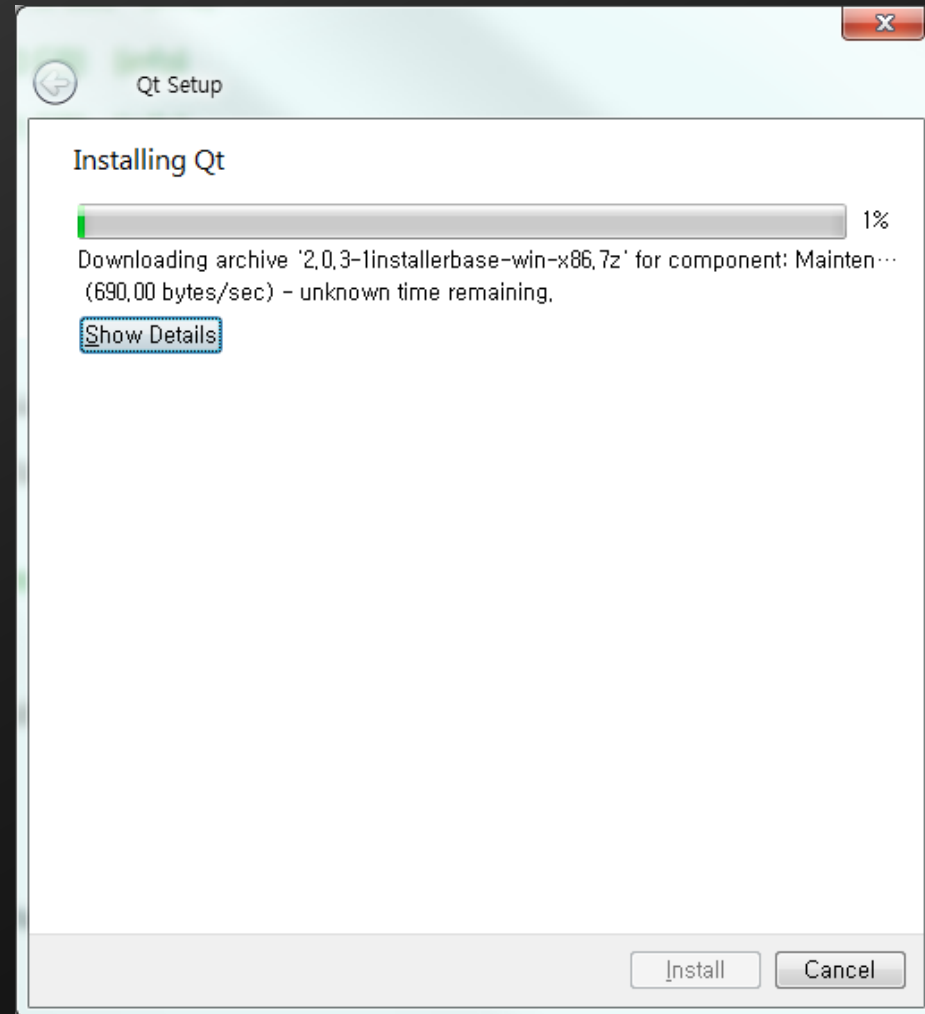
There are a few ways you can get started with Qt. To help us find the option for you, tell us what your application or device will be developed for...

Commercial deployment  In-house deployment, private use, or student use  Open source distribution under an LGPL or GPL license

Get started

# Installation

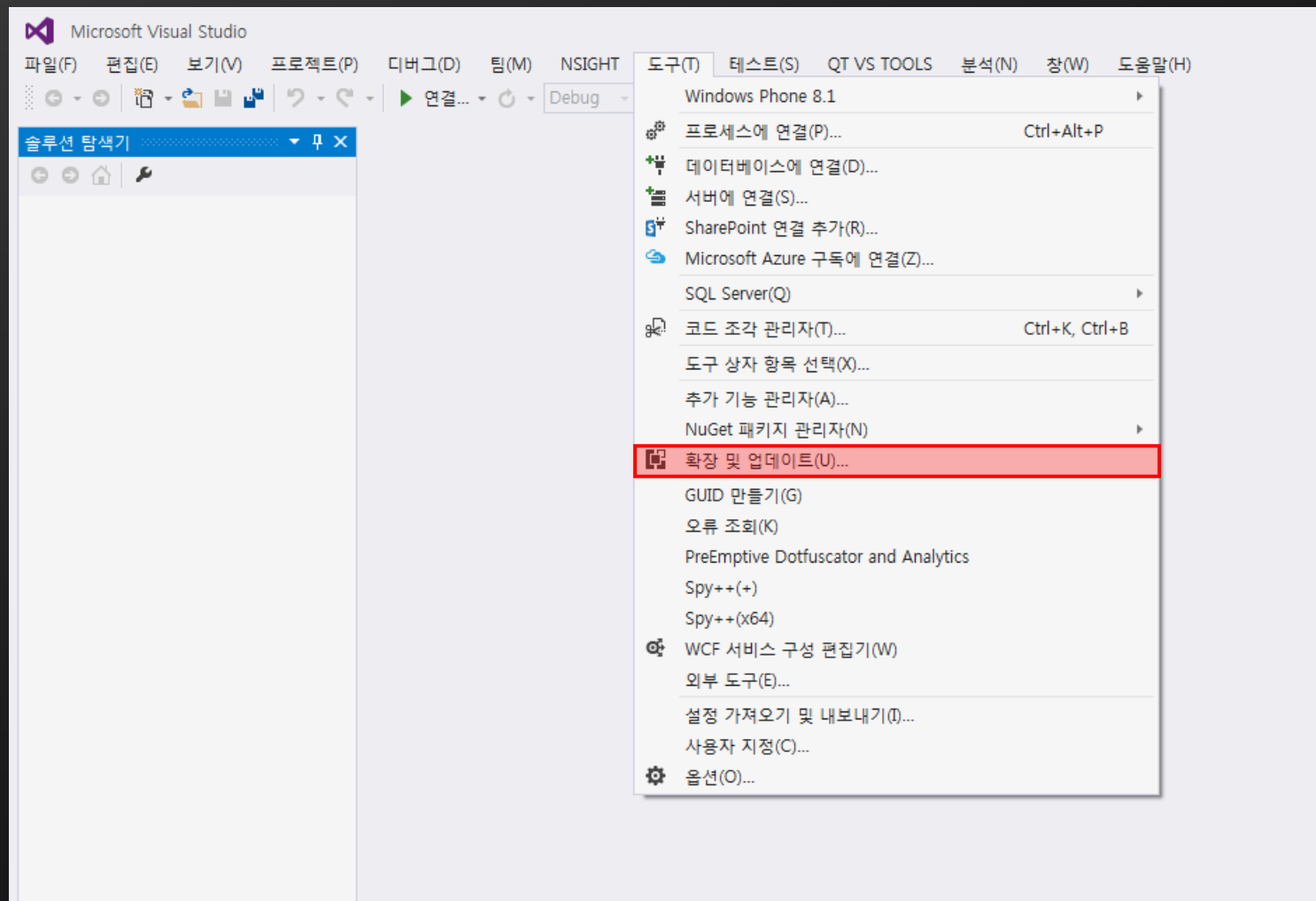
## ❖ Set up Qt





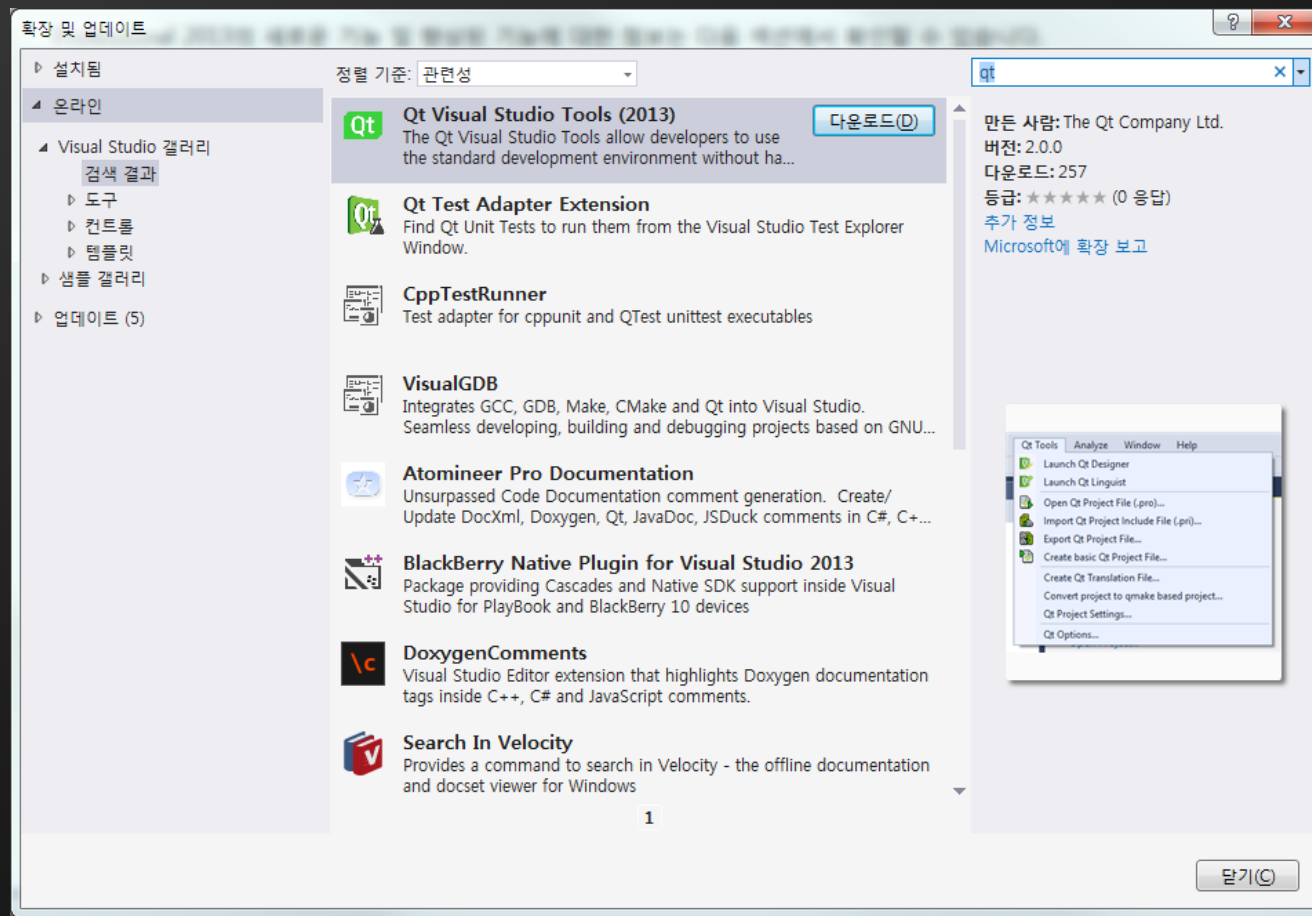
# Installation

## ❖ Visual studio extension



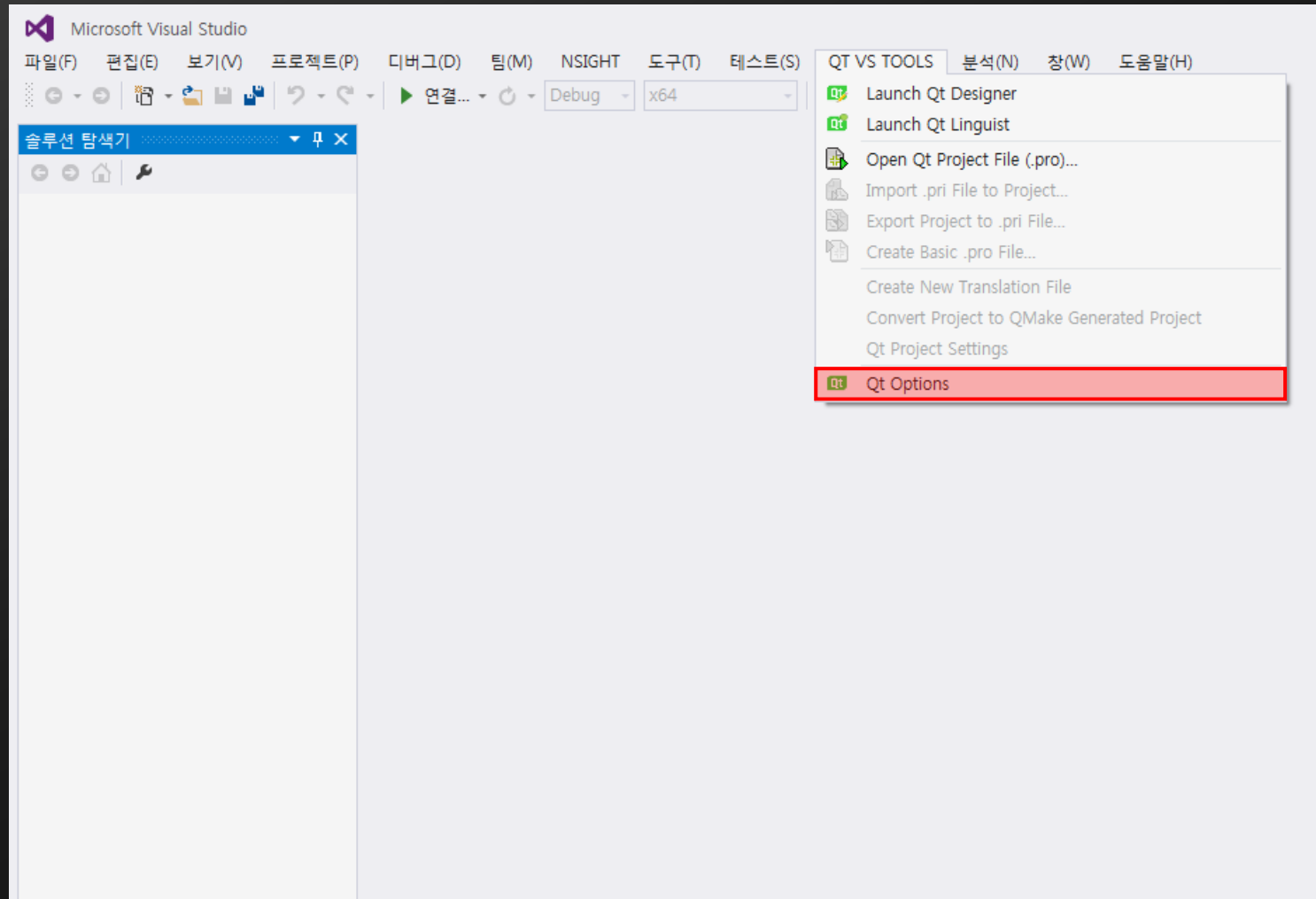
# Installation

## ❖ Visual studio extension



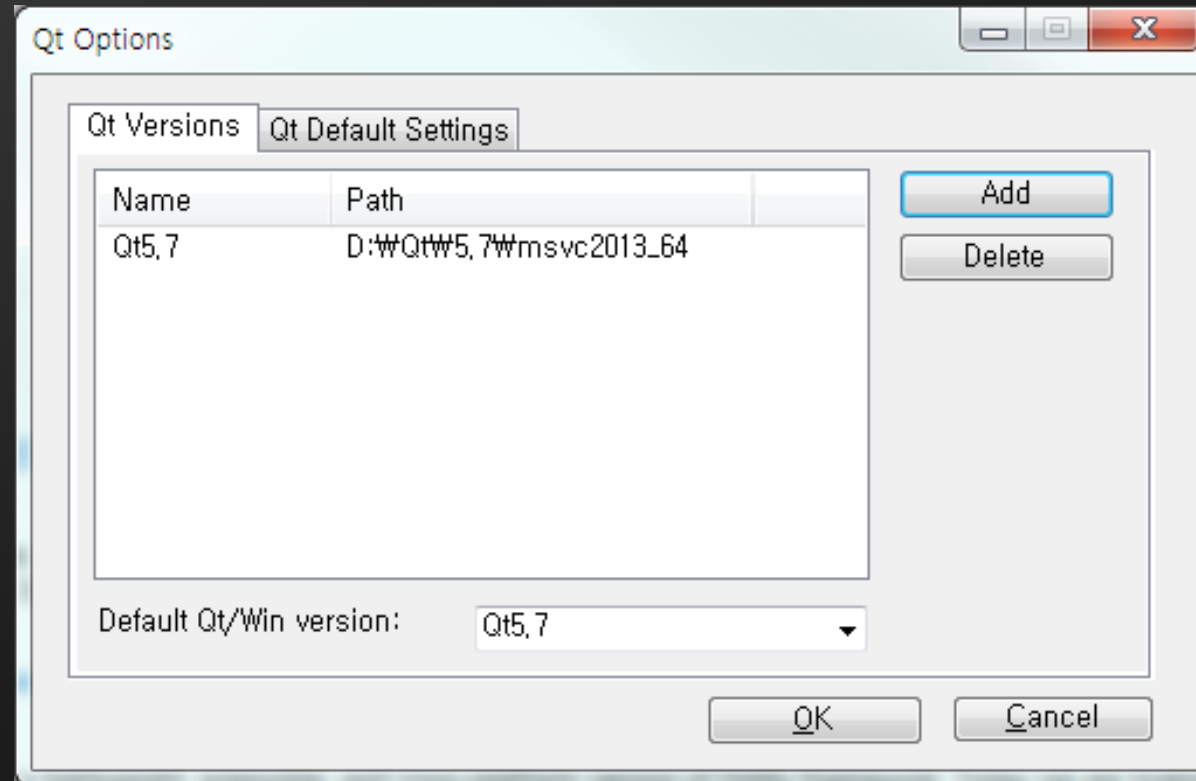
# Installation

## ❖ Visual studio extension



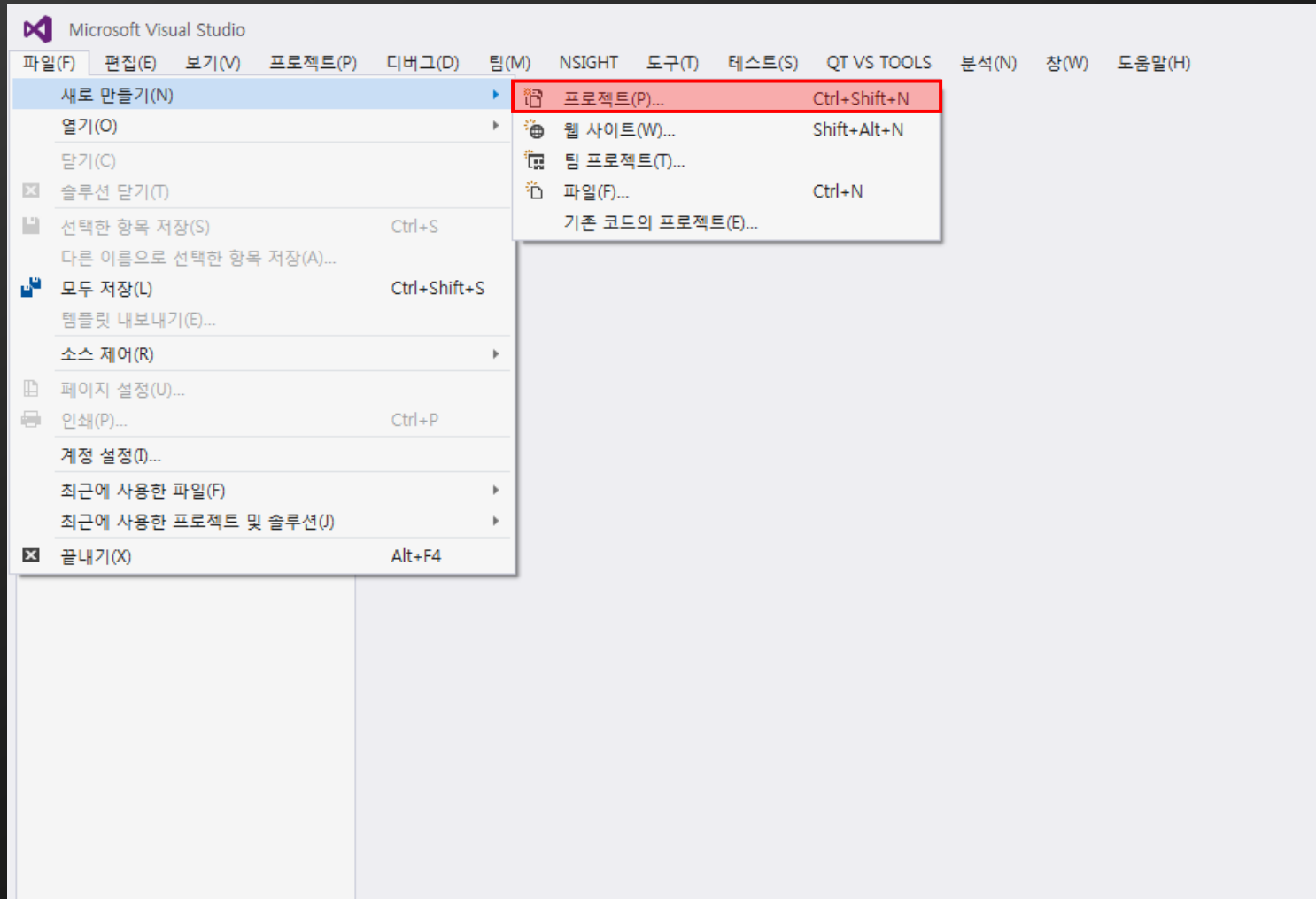
# Installation

## ❖ Visual studio extension



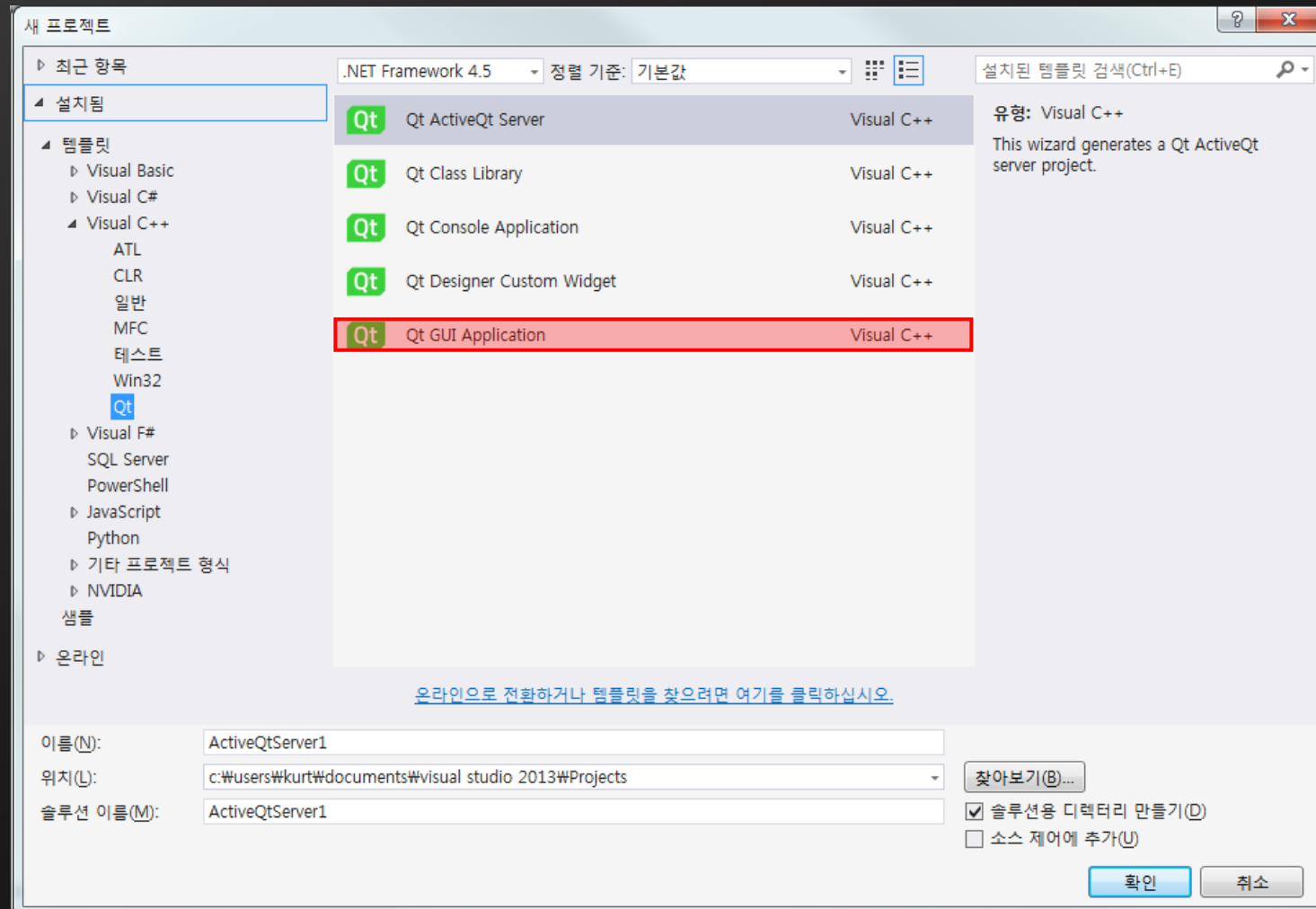
# Qt application

## ❖ Creating a Qt application



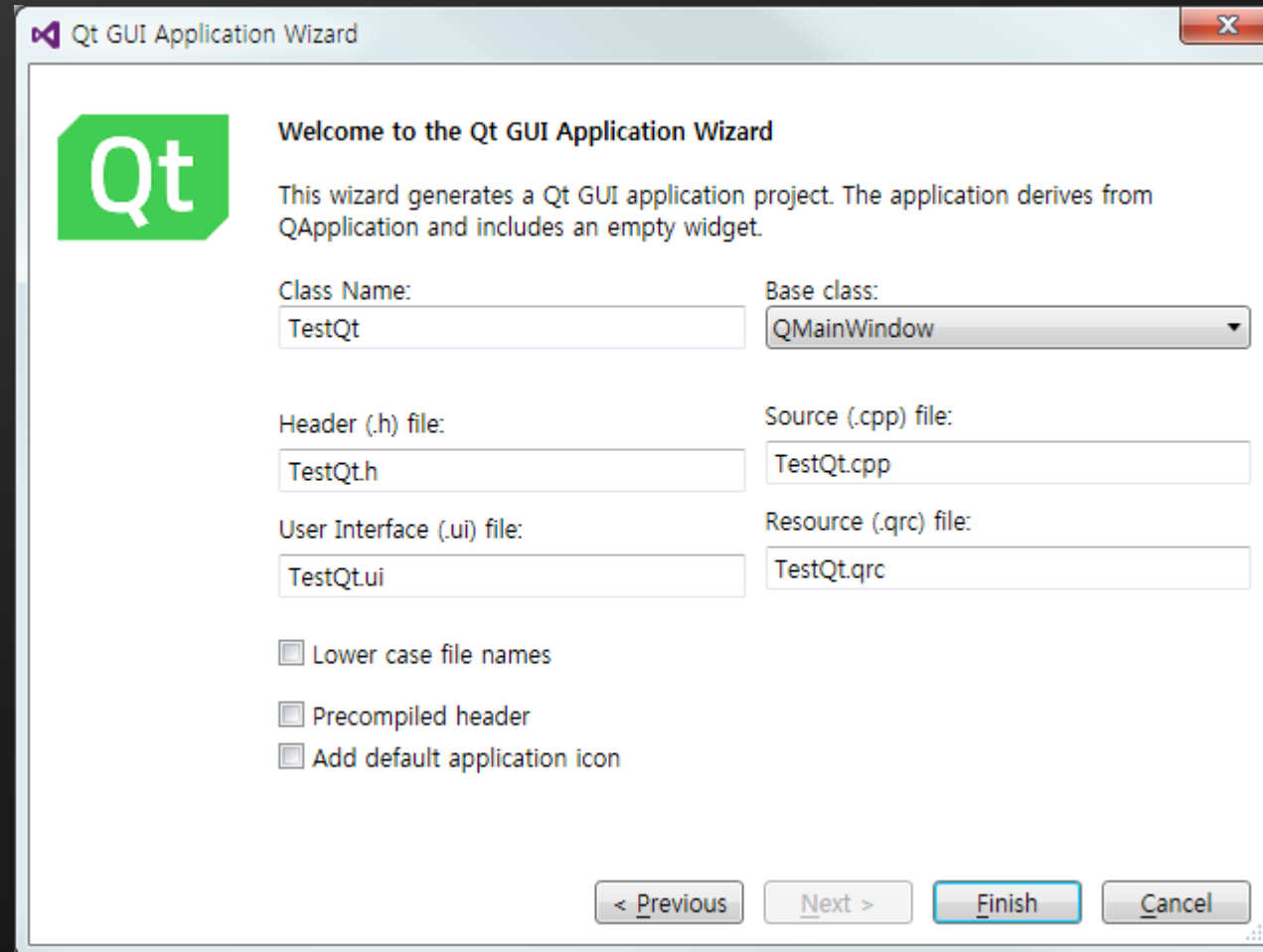
# Qt application

## ❖ Creating a Qt application



# Qt application

## ❖ Creating a Qt application



The screenshot shows the 'Qt GUI Application Wizard' dialog box. It features a Qt logo on the left and a title bar with a close button. The main content area is titled 'Welcome to the Qt GUI Application Wizard' and contains the following fields and options:

- Class Name:** TestQt
- Base class:** QMainWindow
- Header (.h) file:** TestQt.h
- Source (.cpp) file:** TestQt.cpp
- User Interface (.ui) file:** TestQt.ui
- Resource (.qrc) file:** TestQt.qrc

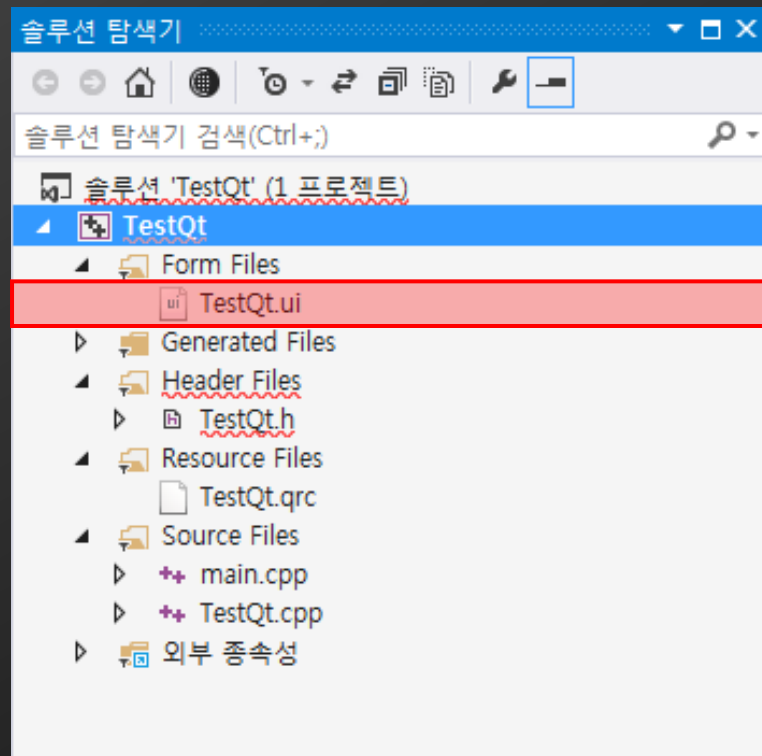
Below the file fields are three checkboxes:

- Lower case file names
- Precompiled header
- Add default application icon

At the bottom of the dialog are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

# Qt application

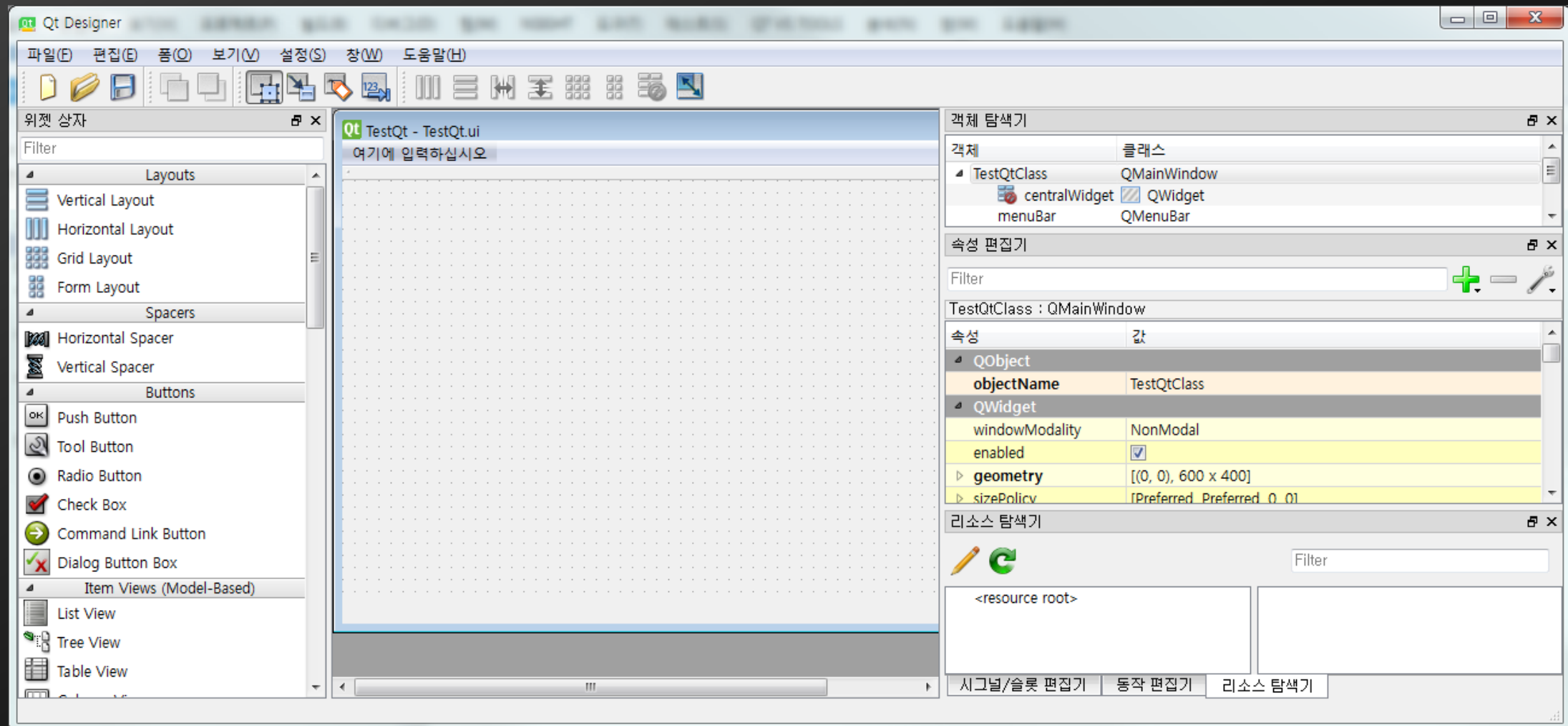
## ❖ Structure of Qt application







# Qt application

## ❖ Structure of Qt application



# Qt application

## ❖ Concept of signal and slot

	
Event(Message)	SIGNAL
Event handler(Function)	SLOT
ON_BN_CLICKED(id, memberFxn)	connect(sender, signal, receiver, member, Qt::ConnectionType = Qt::AutoConnection)

# Qt application

## ❖ Concept of signal and slot

The screenshot shows the Qt Designer interface. On the left, the widget box contains various layout managers (Vertical, Horizontal, Grid, Form), spacers, and buttons. The central canvas displays a dialog window titled "Qt Dialog - untitled+" with a single "PushButton" widget. On the right, the Properties panel shows the configuration for the selected "pushButton : QPushButton".

**Properties Panel Data:**

속성 (Property)	값 (Value)
QObject	
objectName	pushButton
QWidget	
enabled	<input checked="" type="checkbox"/>
geometry	[(20, 40), 75 x 23]
sizePolicy	[Minimum, Fixed, 0, 0]
minimumSize	0 x 0
maximumSize	16777215 x 16777215
sizeIncrement	0 x 0
baseSize	0 x 0
palette	상속됨 (Inherited)
font	A [Gulim, 9]
cursor	화살표 (Arrow)

# Qt application

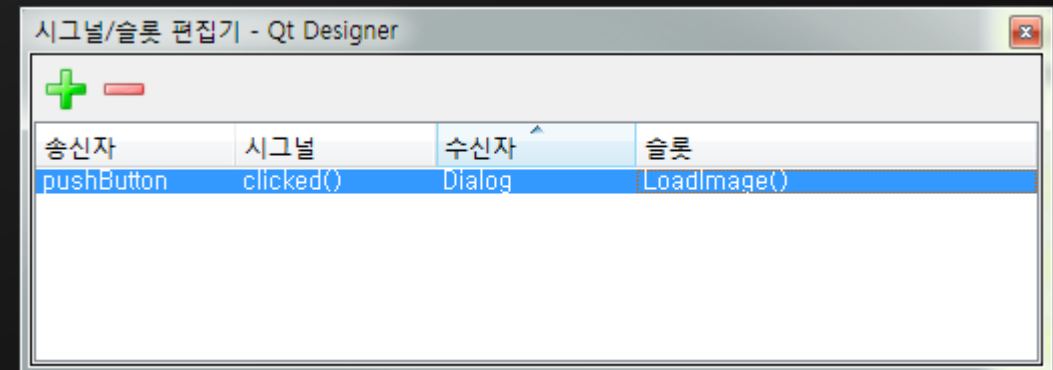
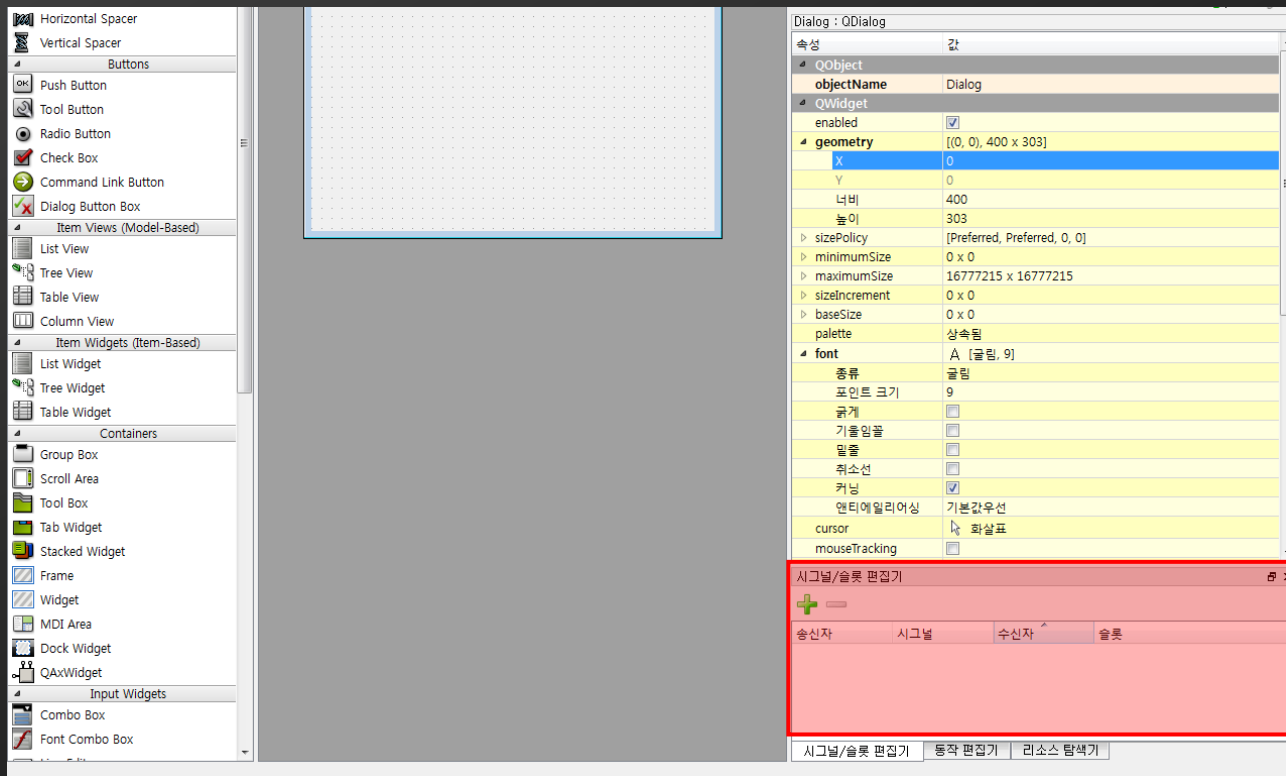
## ❖ Concept of signal and slot

The screenshot shows the Qt Designer interface. In the center, a window titled "Qt Dialog - untitled\*" contains a "Push Button" widget. A red line connects the "clicked()" signal from the button to the "LoadImage()" slot, which is represented by a ground symbol. On the left, the "Widget Box" (위젯 상자) is visible, showing various layout managers, spacers, and buttons. On the right, the "Object Inspector" (객체 탐색기) shows the class hierarchy: Dialog (QDialog) and QPushButton (QPushButton). Below it, the "Property Editor" (속성 편집기) displays the properties for the Dialog widget, including its geometry and size policies.

속성	값
QObject	
objectName	Dialog
QWidget	
enabled	<input checked="" type="checkbox"/>
geometry	[(0, 0), 400 x 303]
X	0
Y	0
너비	400
높이	303
sizePolicy	[Preferred, Preferred, 0, 0]
minimumSize	0 x 0
maximumSize	16777215 x 16777215
sizeIncrement	0 x 0
baseSize	0 x 0

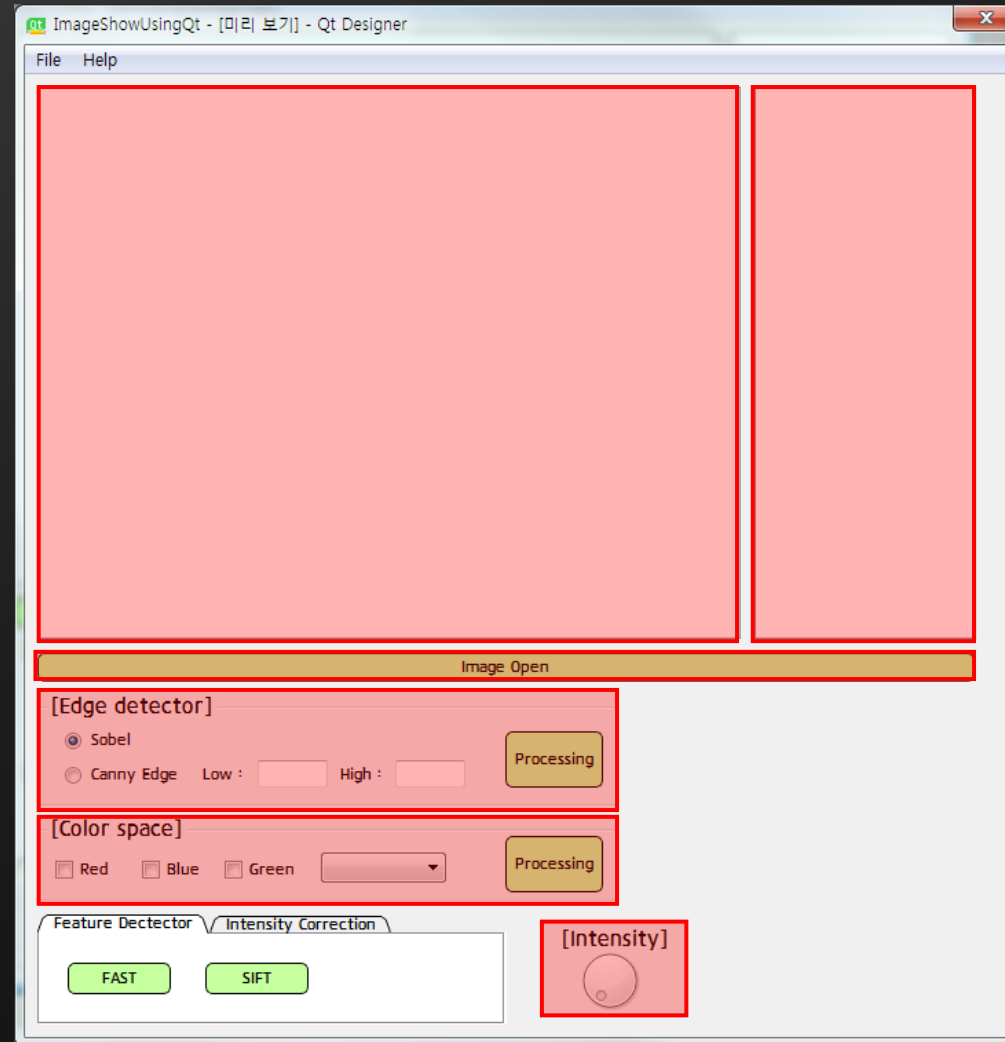
# Qt application

## ❖ Concept of signal and slot



# Qt application

## ❖ ImageShowUsingQt



# Qt application

## ❖ QPushButton & QFileDialog

[ImageShowUsingQt.h]

```
#pragma once

#include <QtWidgets/QMainWindow>
#include "ui_ImageShowUsingQt.h"

class ImageShowUsingQt : public QMainWindow
{
    Q_OBJECT

public:
    ImageShowUsingQt(QWidget *parent = Q_NULLPTR);

private:
    Ui::ImageShowUsingQtClass ui;
};
```

```
#pragma once

#include <QtWidgets/QMainWindow>
#include "ui_ImageShowUsingQt.h"
#include "QFileDialog"

class ImageShowUsingQt : public QMainWindow
{
    Q_OBJECT

public:
    ImageShowUsingQt(QWidget *parent = Q_NULLPTR);
private:
    Ui::ImageShowUsingQtClass ui;
    private slots:
    void LoadImage();
};
```

# Qt application

## ❖ QPushButton & QFileDialog

[ImageShowUsingQt.cpp]

```

#include "ImageShowUsingQt.h"

ImageShowUsingQt::ImageShowUsingQt(QWidget *parent)
    : QMainWindow(parent)
{
    ui.setupUi(this);
}

void ImageShowUsingQt::LoadImage()
{
    ui.listWidget->clear();
    HistoryImg.clear();
    QString qsfileName = QFileDialog::getOpenFileName(this, tr("Open Image"), "../", tr("Image Files (*.png
*.jpg *.bmp)"));
    InputImg = imread(qsfileName.toStdString());
    ShowImage(InputImg);
    HistoryImg.push_back(InputImg);
    ui.listWidget->addItem("Original Image");
}

QString getOpenFileName(QWidget *parent = Q_NULLPTR,
                        const QString &caption = QString(),
                        const QString &dir = QString(),
                        const QString &filter = QString(),
                        QString *selectedFilter = Q_NULLPTR,
                        Options options = Options());

```



# Qt application

## ❖ QPushButton & QFileDialog

[QString & std::string]

QString -> std::string

```
QString qstring;  
std::string stdstring;  
stdstring = qstring.toStdString();
```

std::string -> QString

```
QString qstring;  
std::string stdstring;  
qstring.fromStdString(stdstring);
```

# Qt application

## ❖ QRadioButton & QLineEdit

```
void ImageShowUsingQt::Processing()
{
    //lineedit
    lowthresh = ui.Lowvalue->text().toInt();
    highthresh = ui.Highvalue->text().toInt();

    //flag
    //-1 : default, 0 : sobel, 1 : Canny
    int flag = -1;
    if (ui.SobelButton->isChecked()) flag = 0;
    if (ui.CannyButton->isChecked()) flag = 1;

    if (flag == -1)
    {
        return;
    }
}
```

```
else if (flag == 0)
{
    cvtColor(InputImg, OutputImg, CV_BGR2GRAY);
    Sobel(OutputImg, OutputImg, CV_8UC1, 1, 1, 5);
    cvtColor(OutputImg, OutputImg, CV_GRAY2BGR);
    ShowImage(OutputImg);
    HistoryImg.push_back(OutputImg);
    ui.listWidget->addItem("Sobel");
}

else if (flag = 1)
{
    cvtColor(InputImg, OutputImg, CV_BGR2GRAY);
    Canny(OutputImg, OutputImg, lowthresh, highthresh);
    cvtColor(OutputImg, OutputImg, CV_GRAY2BGR);
    ShowImage(OutputImg);
    HistoryImg.push_back(OutputImg);
    QString th;

    th.sprintf("Canny (low : %d, high : %d)", lowthresh, highthresh);
    ui.listWidget->addItem(th);
}
}
```

}

# Qt application

## ❖ QCheckBox & QComboBox

```

void ImageShowUsingQt::Processing2()
{
    int i = ui.comboBox->currentIndex();
    if (i == 0)
    {
        Mat temp;
        vector<Mat> Channel;
        split(InputImg, Channel);
        if (!ui.RedButton->isChecked())
        {
            Channel[2] = 0;
        }

        if (!ui.BlueButton->isChecked())
        {
            Channel[1] = 0;
        }

        if (!ui.GreenButton->isChecked())
        {
            Channel[0] = 0;
        }

        merge(Channel, temp);
        ShowImage(temp);
    }

    if (i == 1)
    {
        Mat temp;
        cvtColor(InputImg, temp, CV_BGR2GRAY);
        cvtColor(temp, temp, CV_GRAY2BGR);
        ShowImage(temp);
    }
}

```

# Qt application

## ❖ QCheckBox & QComboBox

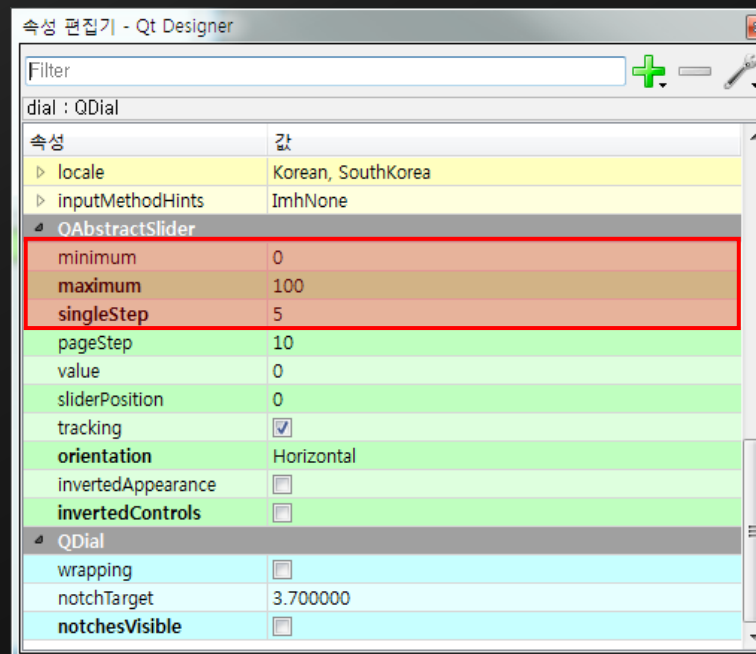
```
void ImageShowUsingQt::Activation()
{
    int i = ui.comboBox->currentIndex();
    if (i == 0)
    {
        ui.RedButton->setCheckable(1);
        ui.BlueButton->setCheckable(1);
        ui.GreenButton->setCheckable(1);
    }

    else
    {
        ui.RedButton->setCheckable(0);
        ui.BlueButton->setCheckable(0);
        ui.GreenButton->setCheckable(0);
    }
}
```

# Qt application

## ❖ Dial

송신자	시그널	수신자	슬롯
ImageOpen	clicked()	ImageShowUsingQtClass	LoadImage()
Process...Button	clicked()	ImageShowUsingQtClass	Processing()
listWidget	itemClicked(QListWidgetItem*)	ImageShowUsingQtClass	ShowHistory()
Process...tton_2	clicked()	ImageShowUsingQtClass	Processing2()
comboBox	currentIndexChanged(QString)	ImageShowUsingQtClass	Activation()
dial	valueChanged(int)	ImageShowUsingQtClass	IntensityControl()



# Qt application

## ❖ Dial

```
void ImageShowUsingQt::IntensityControl()
{
    Mat temp;
    vector<Mat> Channel;
    cvtColor(InputImg, temp, CV_BGR2HSV);
    split(temp, Channel);
    Channel[2]+=ui.dial->value();
    merge(Channel, temp);
    cvtColor(temp, temp, CV_HSV2BGR);
    ShowImage(temp);
}
```

# Qt application

## ❖ QListWidget

송신자	시그널	수신자	슬롯
ImageOpen	clicked()	ImageShowUsingQtClass	LoadImage()
ProcessButton	clicked()	ImageShowUsingQtClass	Processing()
listWidget	itemClicked(QListWidgetItem*)	ImageShowUsingQtClass	ShowHistory()
ProcessButton_2	clicked()	ImageShowUsingQtClass	Processing2()
comboBox	currentIndexChanged(QString)	ImageShowUsingQtClass	Activation()
dial	valueChanged(int)	ImageShowUsingQtClass	IntensityControl()

```
void ImageShowUsingQt::ShowHistory()
{
    int i = ui.listWidget->currentRow();
    ShowImage(HistoryImg[i]);
}
```

*cf. Processing*

```
else if (flag == 0)
{
    cvtColor(InputImg, OutputImg, CV_BGR2GRAY);
    Sobel(OutputImg, OutputImg, CV_8UC1, 1, 1, 5);
    cvtColor(OutputImg, OutputImg, CV_GRAY2BGR);
    ShowImage(OutputImg);
    HistoryImg.push_back(OutputImg);
    ui.listWidget->addItem("Sobel");
}
```

# Qt application

## ❖ QGraphicsView

```
void ImageShowUsingQt::ShowImage(Mat image)
{
    image.copyTo(LastImg);
    scene.clear();
    QImage qimg(image.data, image.cols, image.rows, QImage::Format_RGB888);
    scene.addPixmap(QPixmap::fromImage(qimg.rgbSwapped()));

    ui.graphicsView->setScene(&scene);
    ui.graphicsView->show();
}
```



# Qt application

## ❖ ImageShowUsingQt - Demo



Q & A *Thank You!!!*

# Installation

## ❖ Download Qt

Browser

← → ↻ <https://www.qt.io/>

# Get Started with Qt

Determine which option is best for you.

What kind of product are you developing?

Embedded device  Desktop/multiscreen application  Mobile app  All of the above

Start for Free - Qt for Application Development

**Click**

Get started

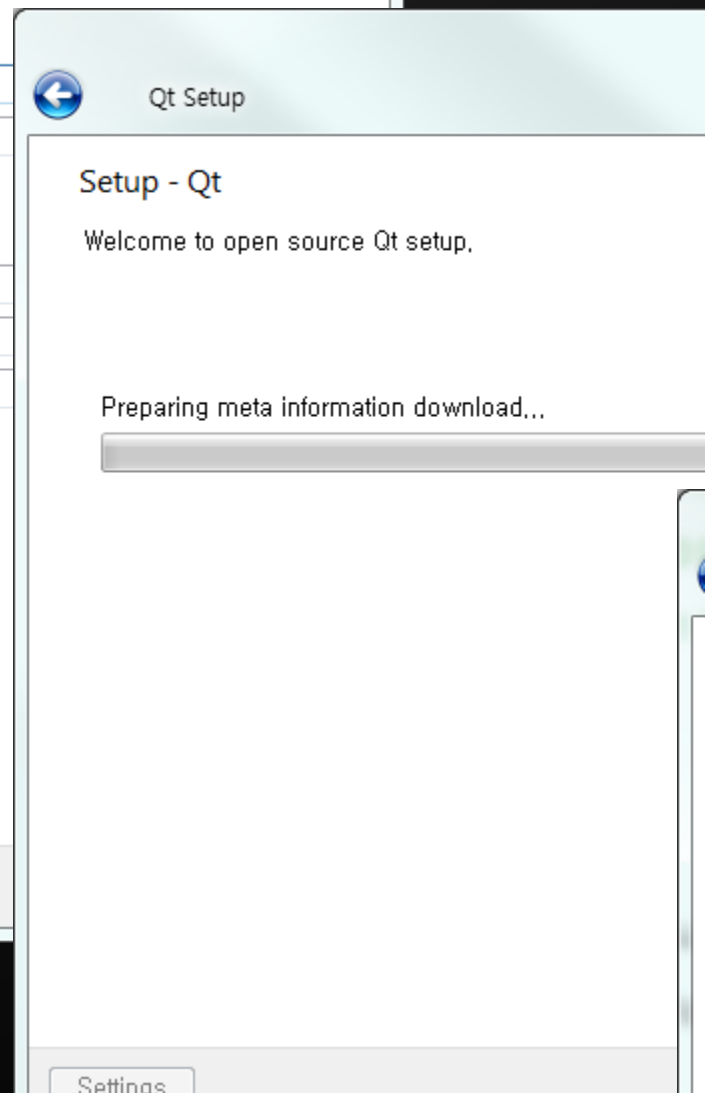
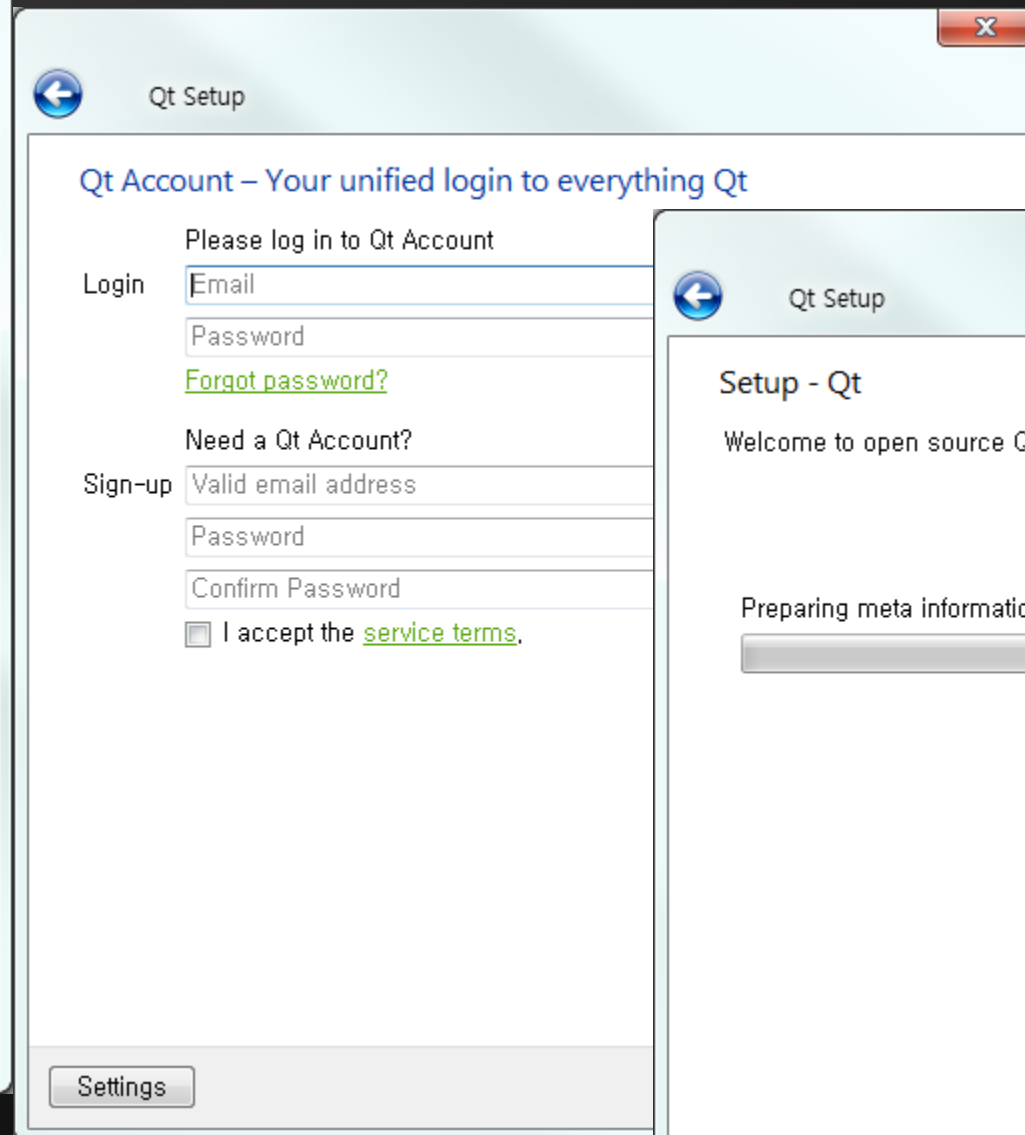
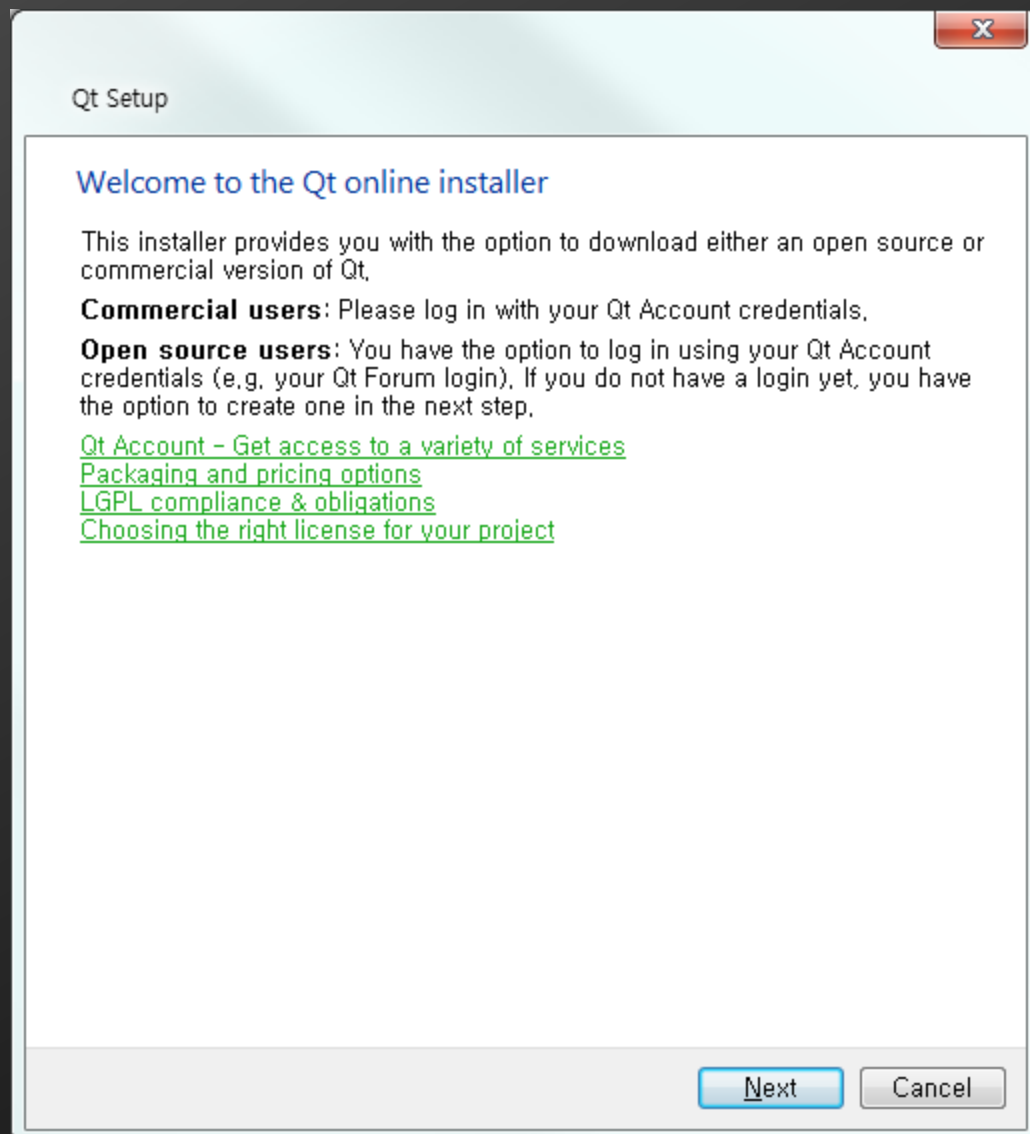
Start again

# h with Qt

ter  
devices.

There a

Comm



기준: 이름: 오름차순

**Application Insights**  
Visual Studio에서 바로 App...  
하여 응용 프로그램을 세부

**Behaviors SDK (XAML)**  
The Behaviors SDK (XAML)  
Windows Store applicatio

**Ceemle OpenCV for**  
OpenCV project template f

**Image Watch**  
Provides a watch window f  
when debugging native C+

**Microsoft Advertising**  
This service allows you to u  
by your Microsoft Advertisi

**Microsoft Advertising**  
This SDK provides a contr  
apps). The control displays

**Microsoft Advertising**  
This SDK provides a contr  
apps). The control displays

**Microsoft ASP.NET ar**  
Provides the latest Web De

Microsoft 해킹과 보안

확장 및 업데이트

정렬 기준: **관련성**

qt

설치됨

온라인

- Visual Studio 갤러리
  - 검색 결과
  - 도구
  - 컨트롤
  - 템플릿
  - 샘플 갤러리
  - 업데이트 (5)

**Qt** **Qt Visual Studio Tools (2013)** [다운로드\(D\)](#)  
The Qt Visual Studio Tools allow developers to use the standard development environment without ha...

**Qt** **Qt Test Adapter Extension**  
Find Qt Unit Tests to run them from the Visual Studio Test Explorer Window.

**CppTestRunner**  
Test adapter for cppunit and QTest unittest executables

**VisualGDB**  
Integrates GCC, GDB, Make, CMake and Qt into Visual Studio. Seamless developing, building and debugging projects based on GNU...

**Atomineer Pro Documentation**  
Unsurpassed Code Documentation comment generation. Create/Update DocXml, Doxygen, Qt, JavaDoc, JSDuck comments in C#, C+...

**BlackBerry Native Plugin for Visual Studio 2013**  
Package providing Cascades and Native SDK support inside Visual Studio for PlayBook and BlackBerry 10 devices

**DoxygenComments**  
Visual Studio Editor extension that highlights Doxygen documentation tags inside C++, C# and JavaScript comments.

**Search In Velocity**  
Provides a command to search in Velocity - the offline documentation and docset viewer for Windows

1

만든 사람: The Qt Company Ltd.  
버전: 2.0.0  
다운로드: 257  
등급: ★★★★★ (0 응답)  
[추가 정보](#)  
[Microsoft에 확장 보고](#)

Qt Tools Analyze Window Help

- Launch Qt Designer
- Launch Qt Linguist
- Open Qt Project File (.pro)...
- Import Qt Project Include File (.pri)...
- Export Qt Project File...
- Create basic Qt Project File...
- Create Qt Translation File...
- Convert project to qmake based project...
- Qt Project Settings...
- Qt Options...

닫기(C)

Qt Options

Qt Versions Qt Def

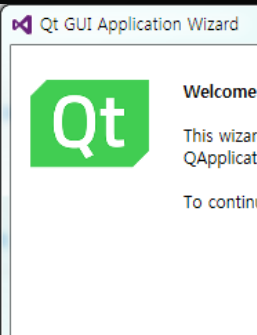
Name

Default Qt/Win ve

Qt G

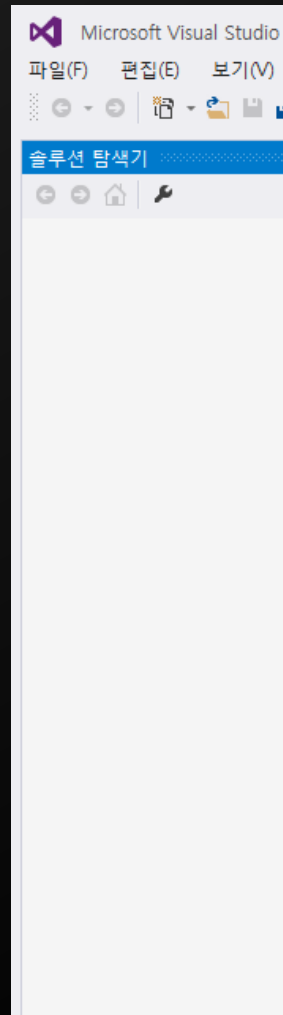
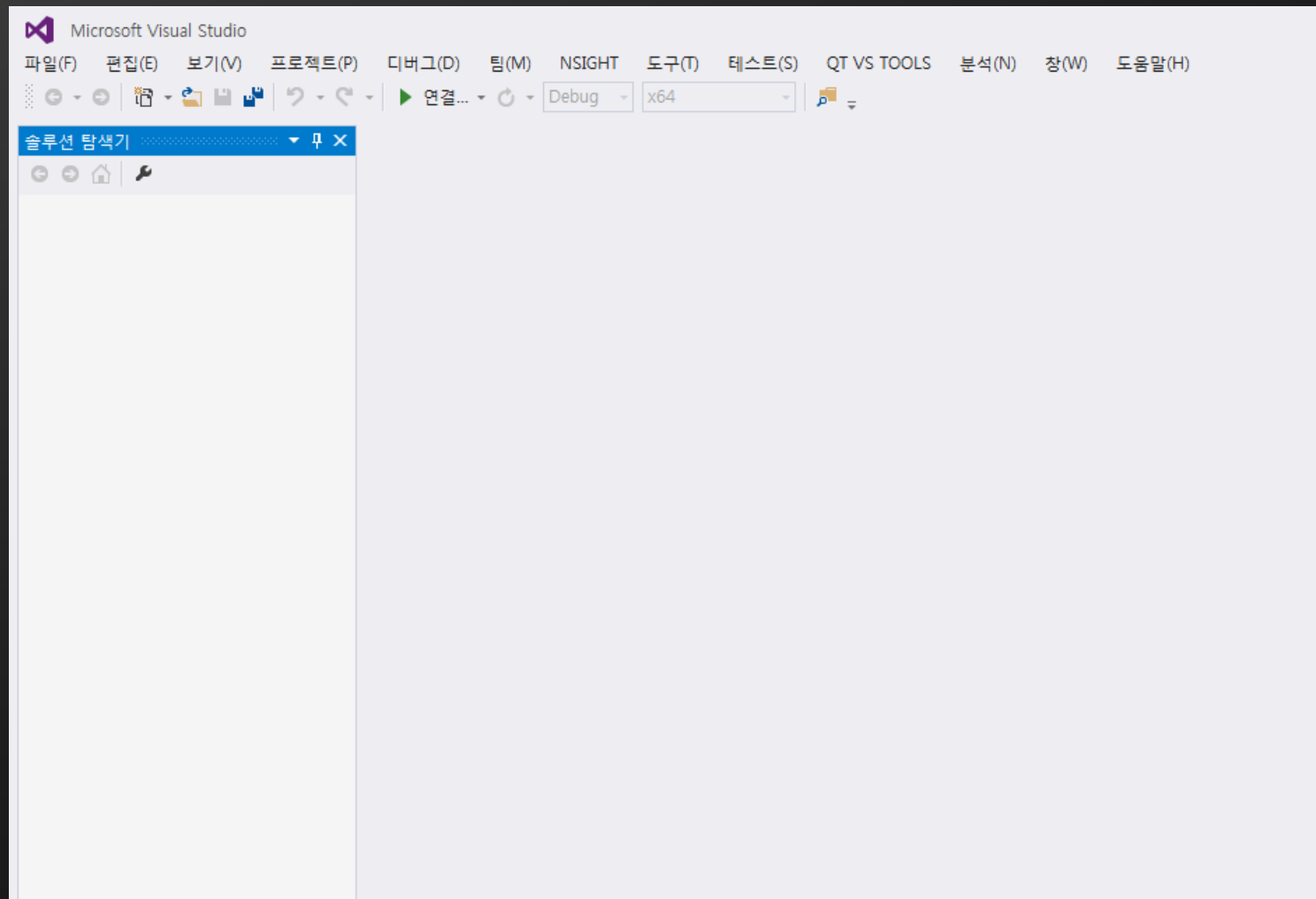
# Installation

## ❖ Visual studio extension



# Installation

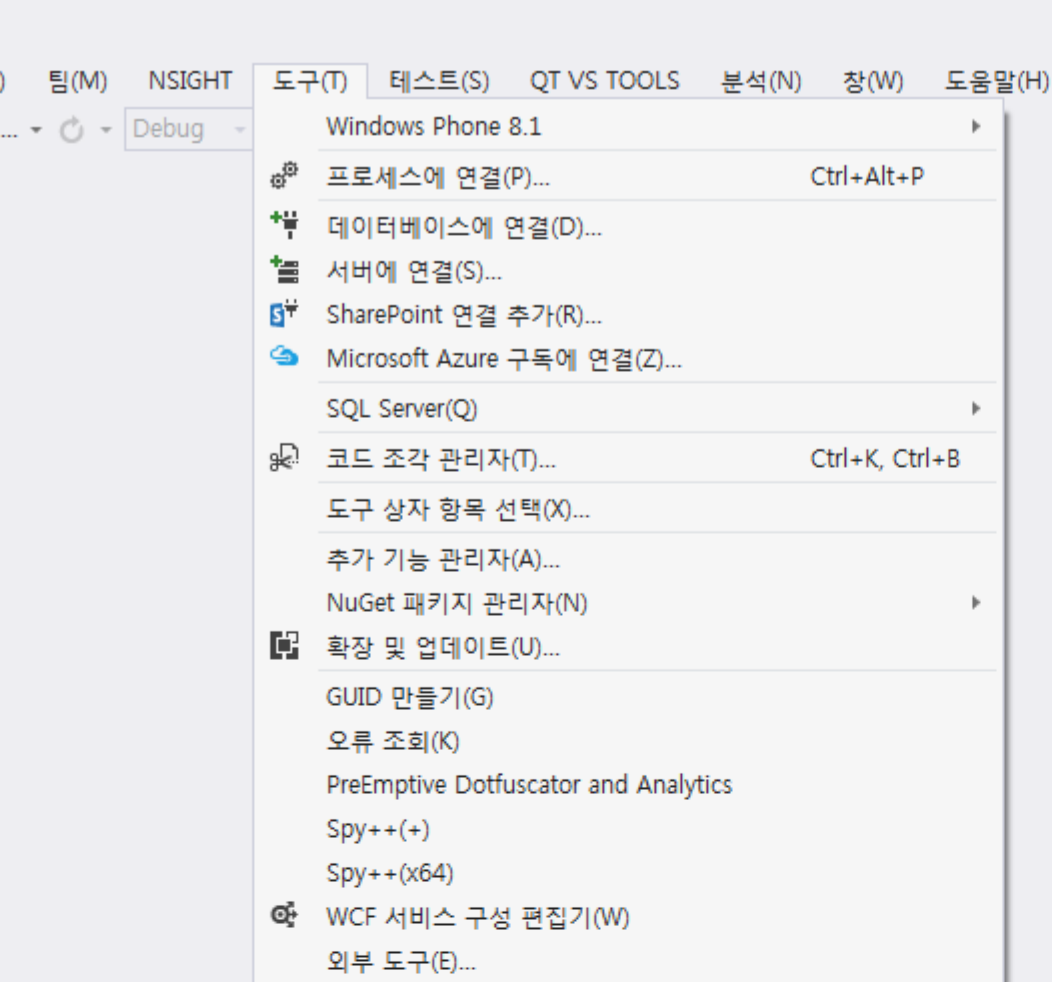
## ❖ Visual studio extension





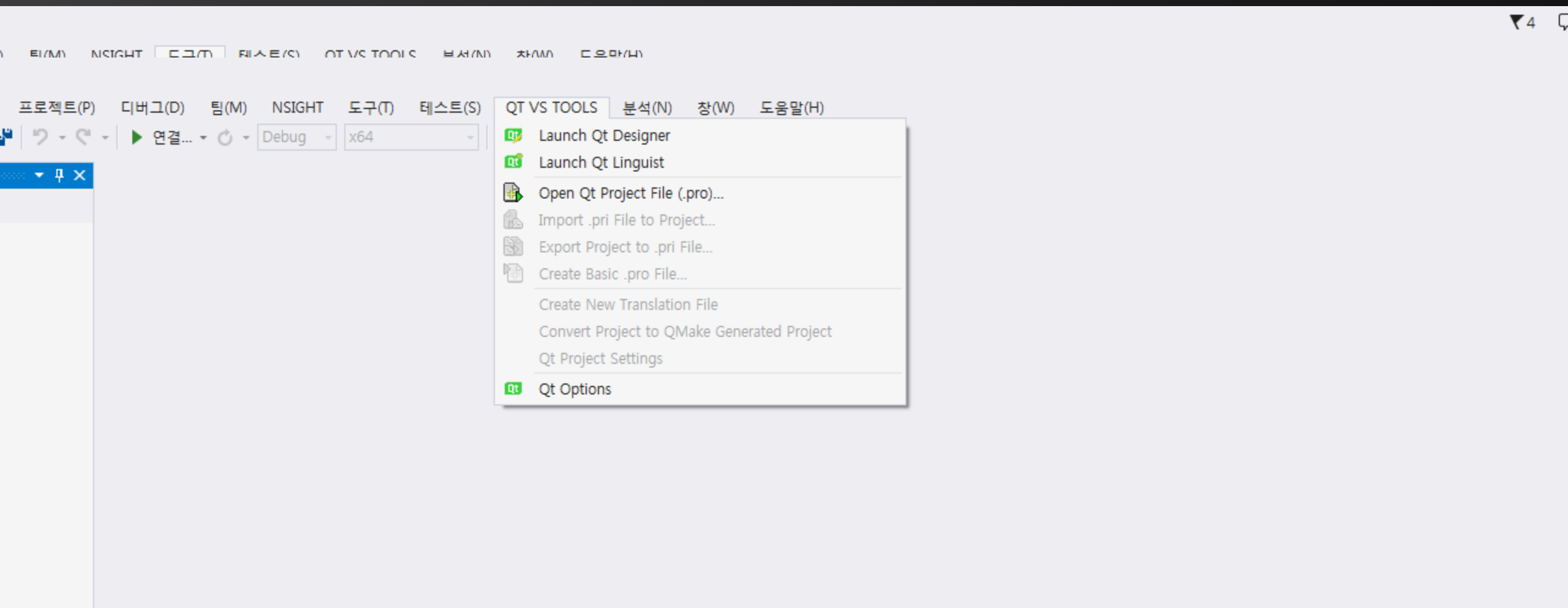
# Installation

## ❖ Visual studio extension



# Installation

## ❖ Visual studio extension



# Qt application

## ❖ QPushButton

[ProjectName.h]

```
#pragma once

#include <QtWidgets/QMainWindow>
#include "ui_ImageShowUsingQt.h"

class ImageShowUsingQt : public QMainWindow
{
    Q_OBJECT

public:
    ImageShowUsingQt(QWidget *parent = Q_NULLPTR);

private:
    Ui::ImageShowUsingQtClass ui;
};
```

```
#pragma once

#include <QtWidgets/QMainWindow>
#include "ui_ImageShowUsingQt.h"
#include "QFileDialog"

class ImageShowUsingQt : public QMainWindow
{
    Q_OBJECT

public:
    ImageShowUsingQt(QWidget *parent = Q_NULLPTR);
private:
    Ui::ImageShowUsingQtClass ui;
    private slots:
    void LoadImage();
};
```