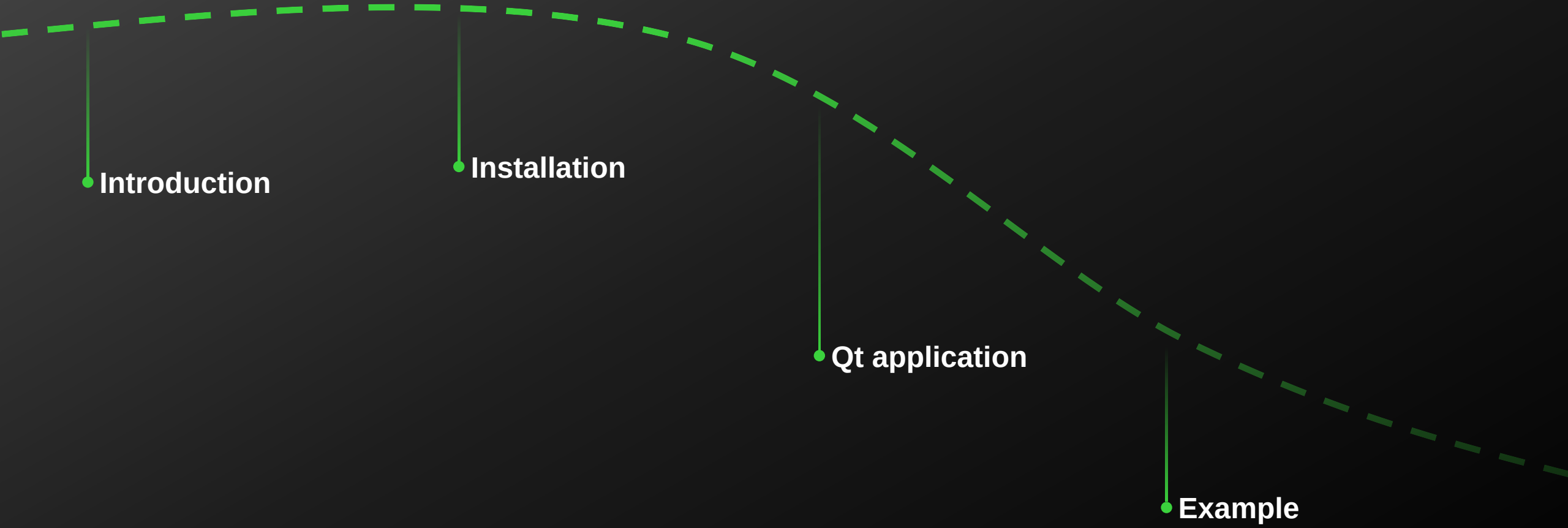
"Write once, compile anywhere"

# Basic Qt Programming Tutorial

Han Sol Kang

ISL Image System Laboratory

# Contents



# Introduction

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

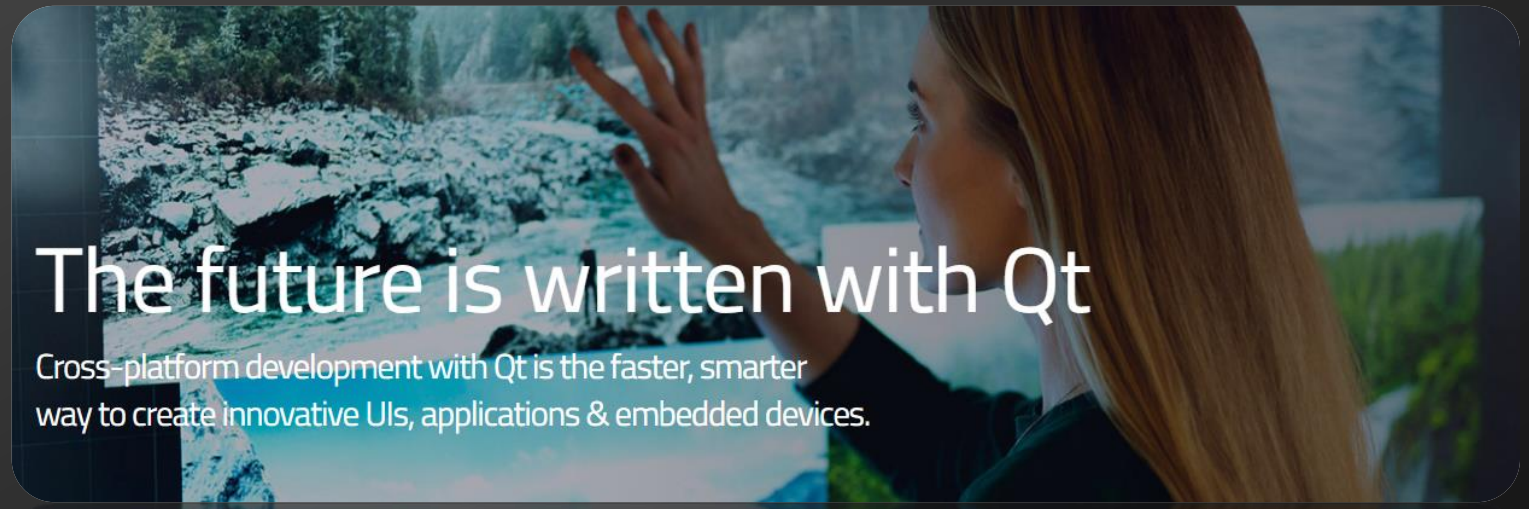
"Write once, compile anywhere" (WOCA)

"Q" Haavard's Emacs typeface

"t" Inspired by Xt (X toolkit)

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

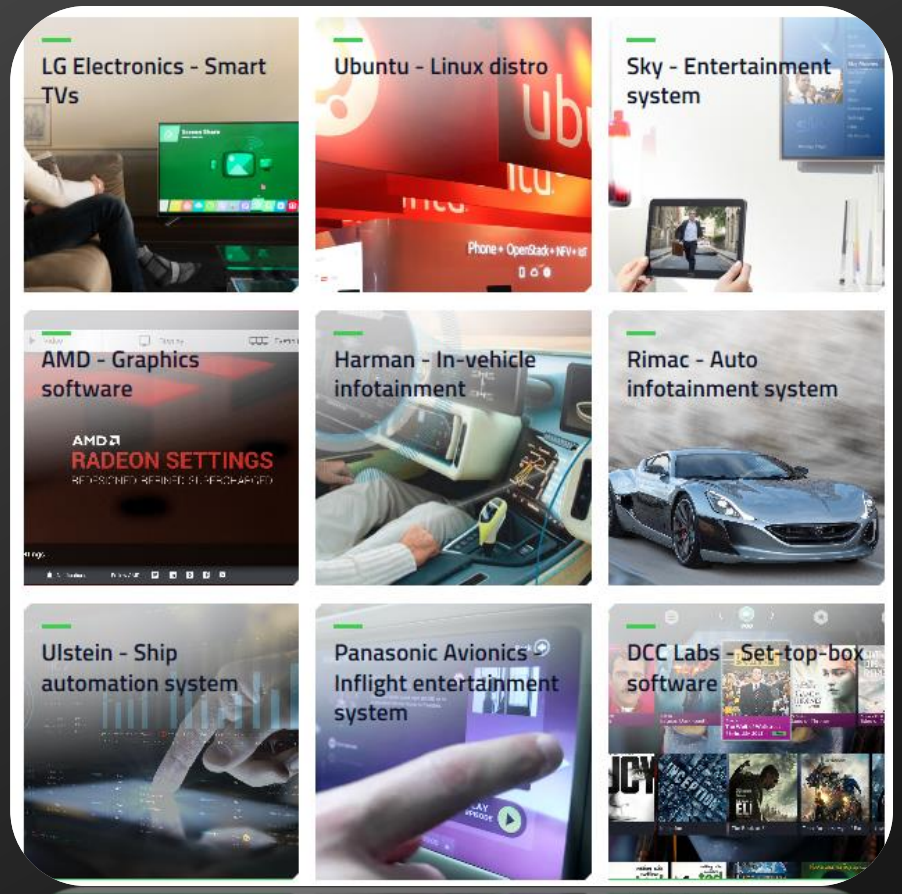
# Introduction




**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)

| Windows | |
|---|---|
| Windows 10 (64-bit) | **MSVC 2015** |
| Windows 10 (32-bit) | MSVC 2015 |
| Windows 8.1 (64-bit) | MSVC 2015, **MSVC 2013**, MinGW 5.3 , MinGW 4.9, MinGW 4.8 |
| Windows 8.1 (32-bit) | MSVC 2015, MSVC 2013, MinGW 5.3 , MinGW 4.9, MinGW 4.8 |
| Windows 7 (64-bit) | MSVC 2015, MSVC 2013, MinGW 5.3 , MinGW 4.9, MinGW 4.8 |
| Windows 7 (32-bit) | MSVC 2015, MSVC 2013, **MinGW 5.3** , MinGW 4.9, MinGW 4.8 |

| Linux/X11 | |
|---|---|
| openSUSE 13.1 (64-bit) | **GCC 4.8.1** |
| Red Hat Enterprise Linux 6.6 (64-bit) | **GCC 4.9.1** |
| Ubuntu 14.04 (64-bit) | **GCC 4.6.3** |
| (Linux 32/64-bit) | GCC 4.8.1, GCC 4.9.1 |

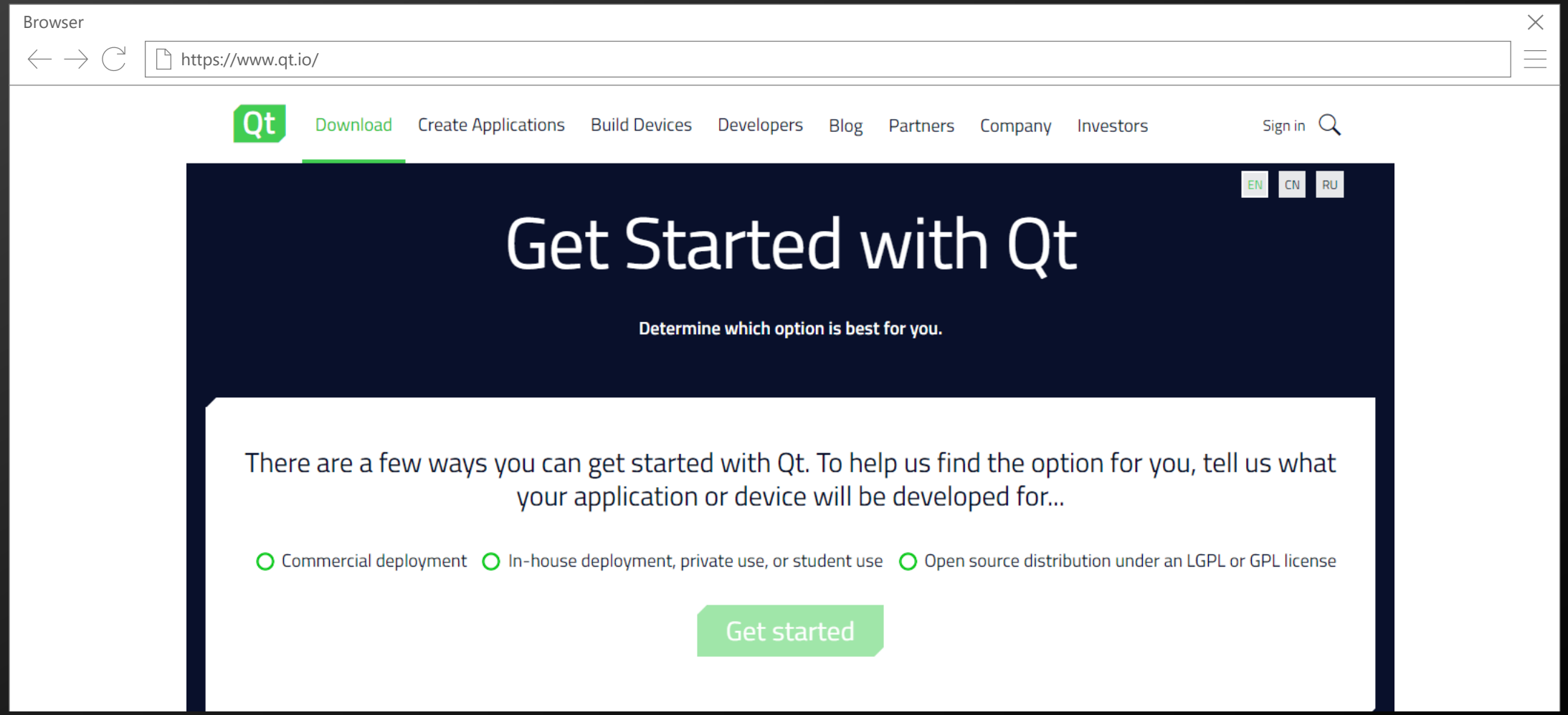
| OS X | |
|---|---|
| OS X **10.8**, **10.9**, **10.10**, **10.11** | **Clang as provided by Apple** |

| **Embedded Platforms: Embedded Linux, QNX** | |
|---|---|
| Embedded Linux | GCC |
| QNX 6.6.0 (armv7le and x86) | **As provided by QNX** |

| **Mobile Platforms: Android, iOS, WinRT** | |
|---|---|
| Windows Phone 8.1 (arm) | **MSVC 2013** |
| Windows Runtime (x86, **x86_64**, arm) | MSVC 2015, **MSVC 2013** |
| iOS 6 and above | **Clang as provided by Apple** |
| Android (API Level: 16) | **GCC as provided by Google** |

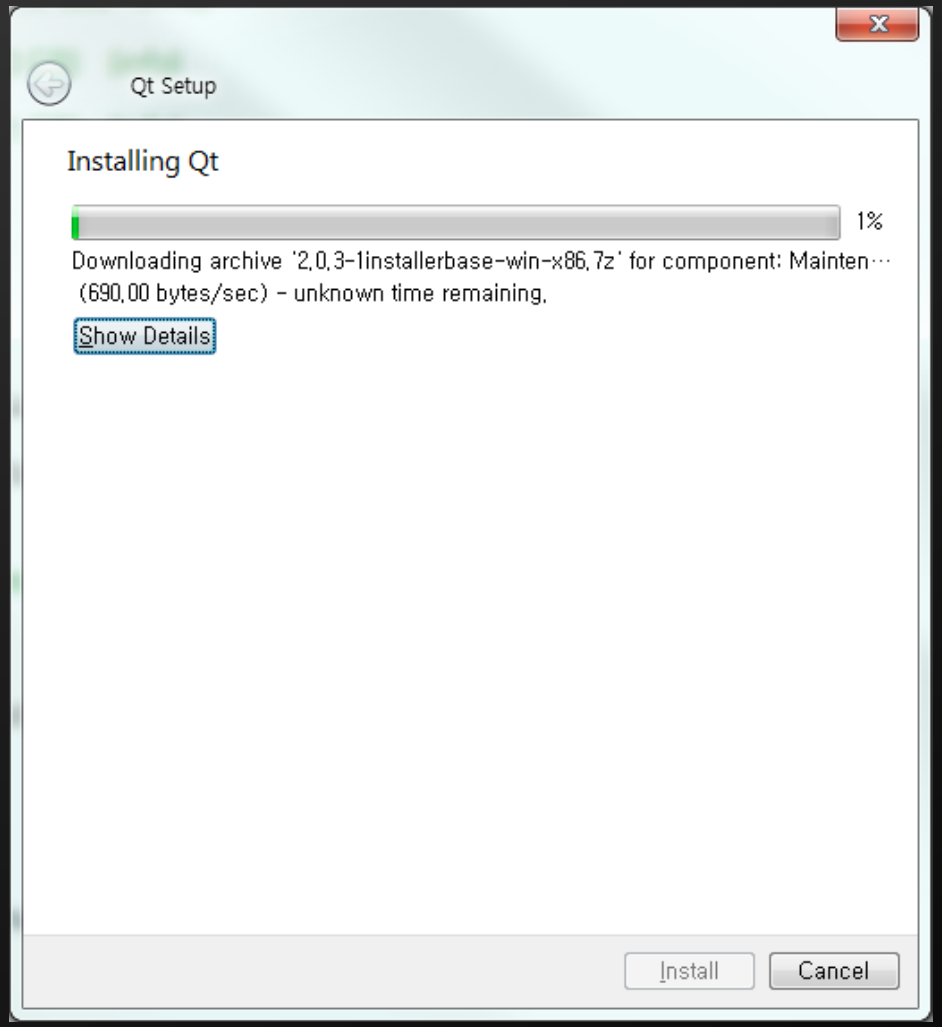
# Installation

- Download Qt



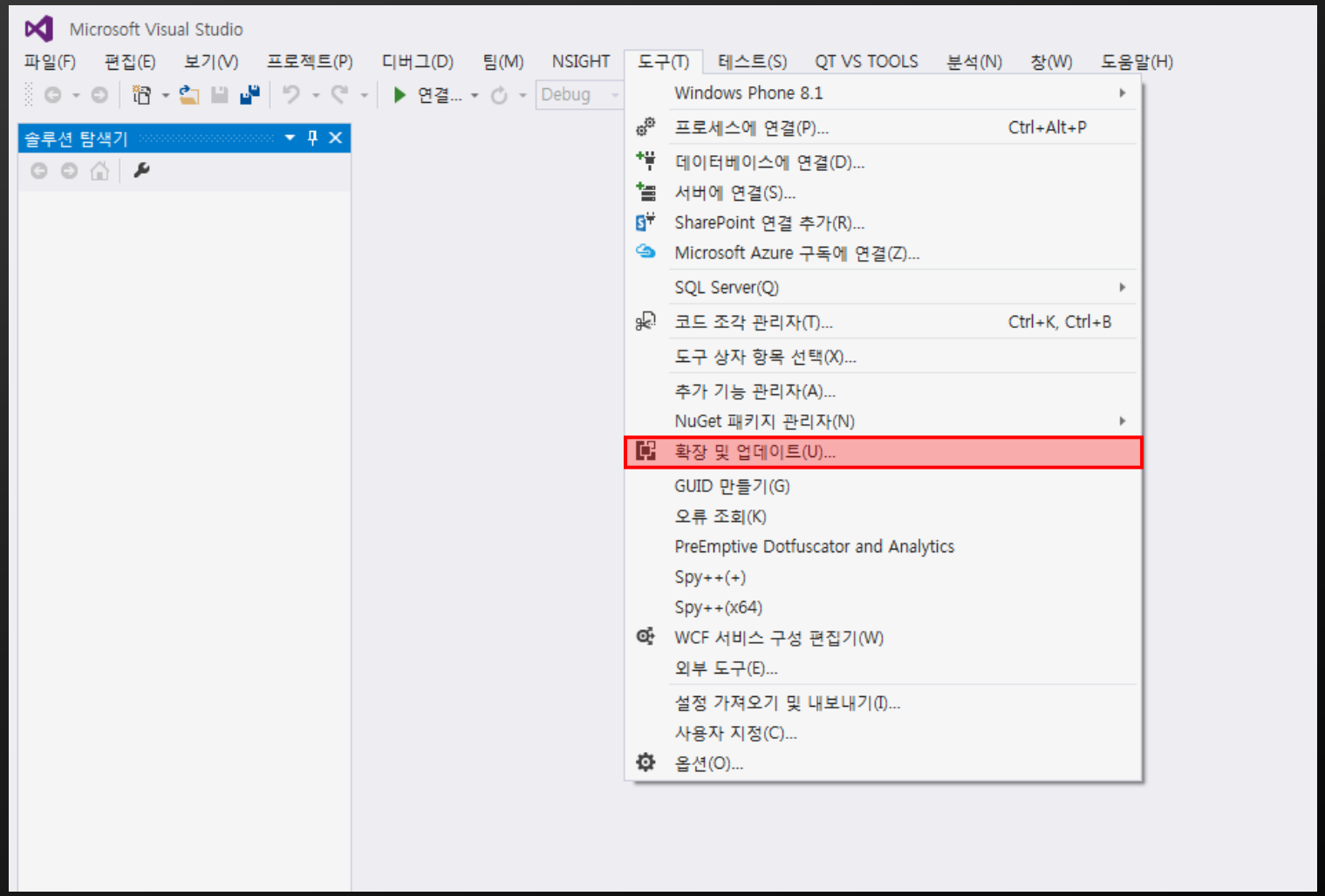
# Installation

- Set up Qt



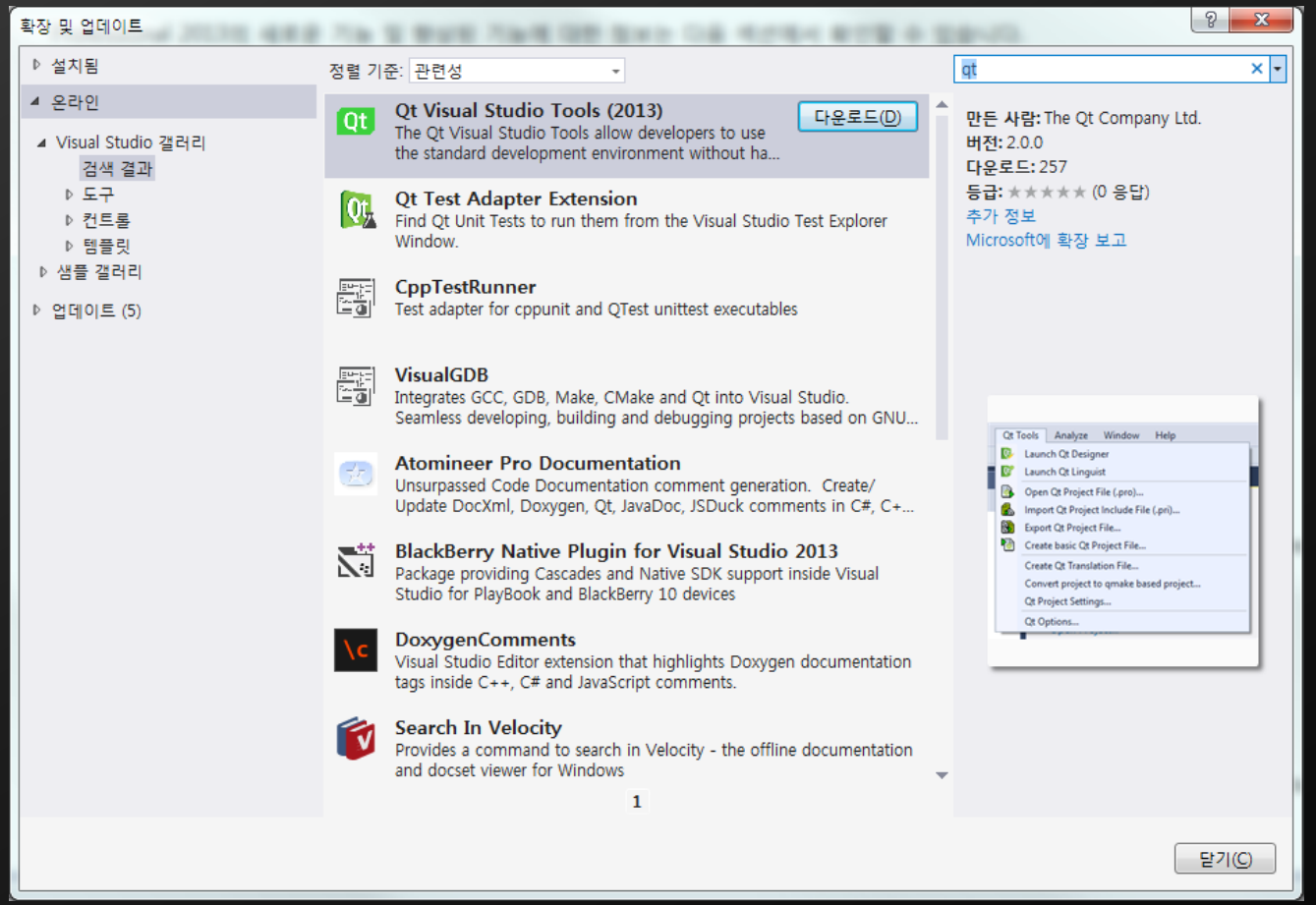
# Installation

❖ Visual studio extension



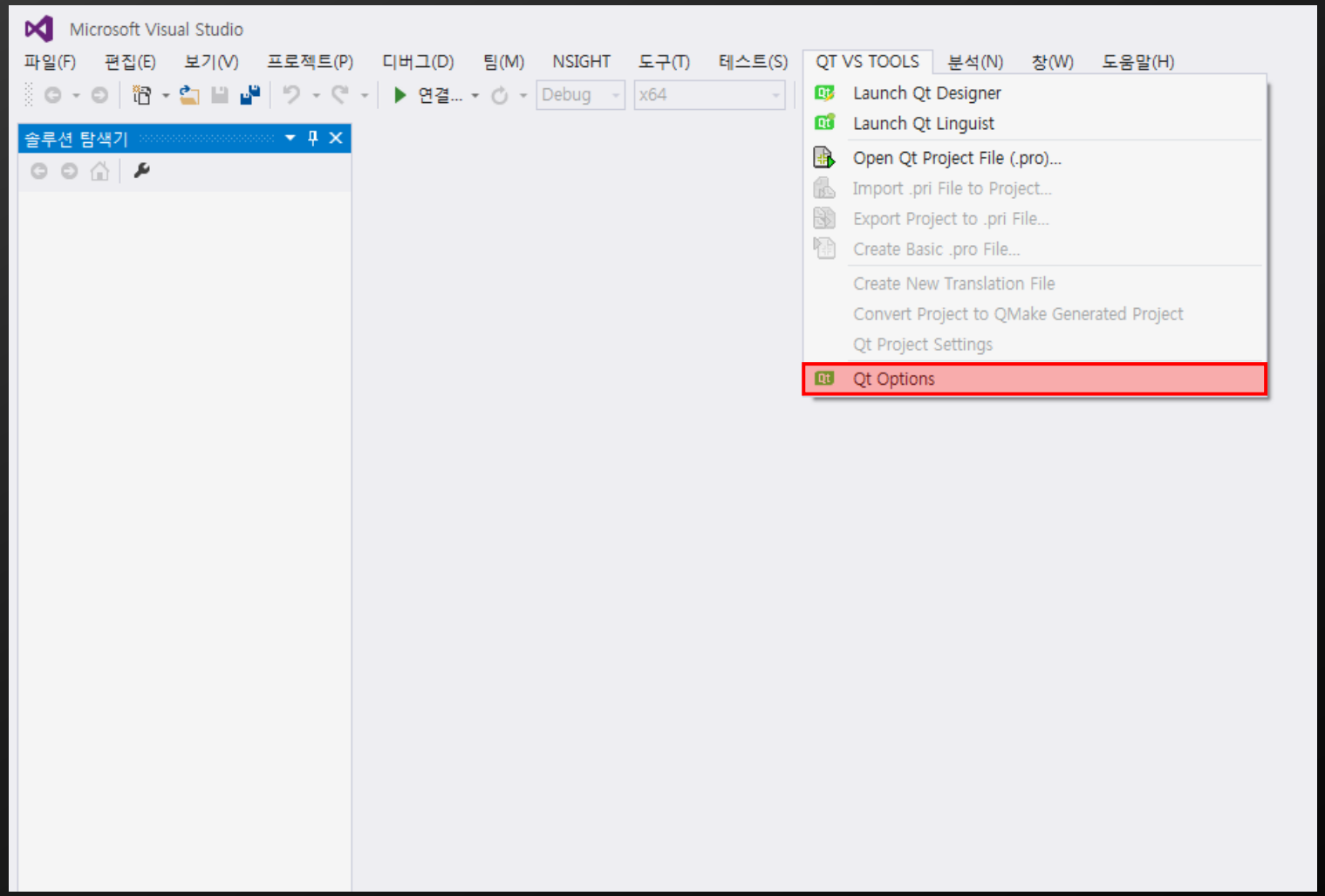
# Installation

- Visual studio extension



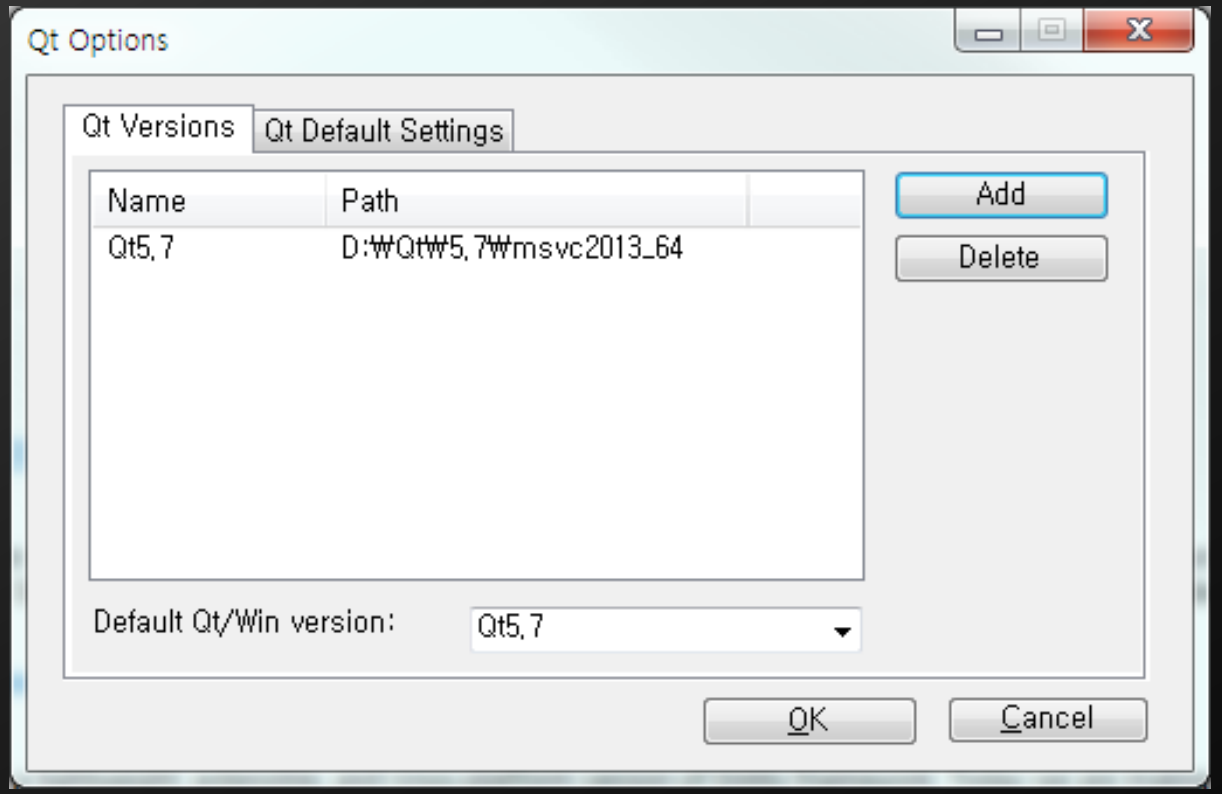
# Installation

❖ Visual studio extension



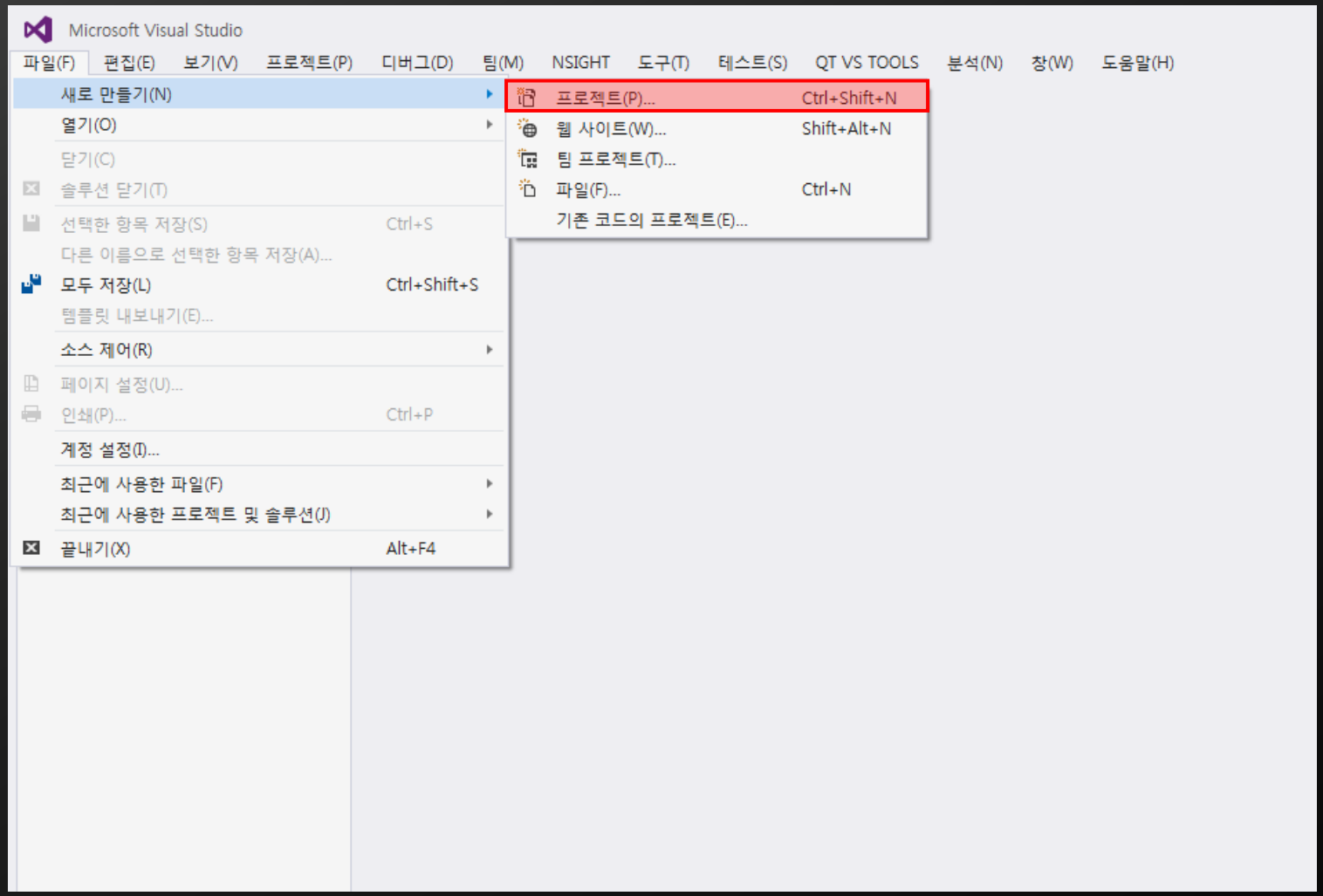
# Installation

- Visual studio extension



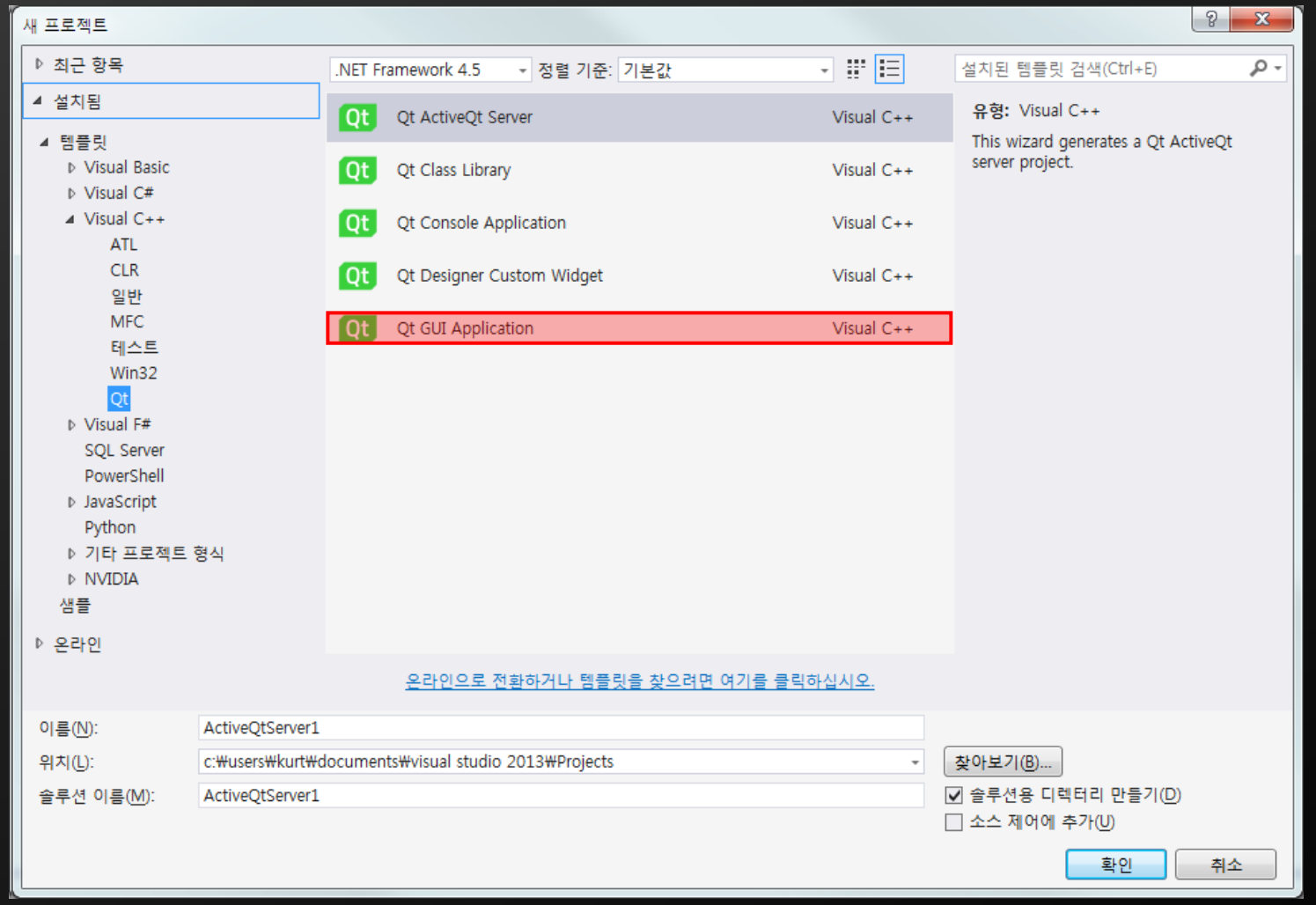
# Qt application

❖ Creating a Qt application



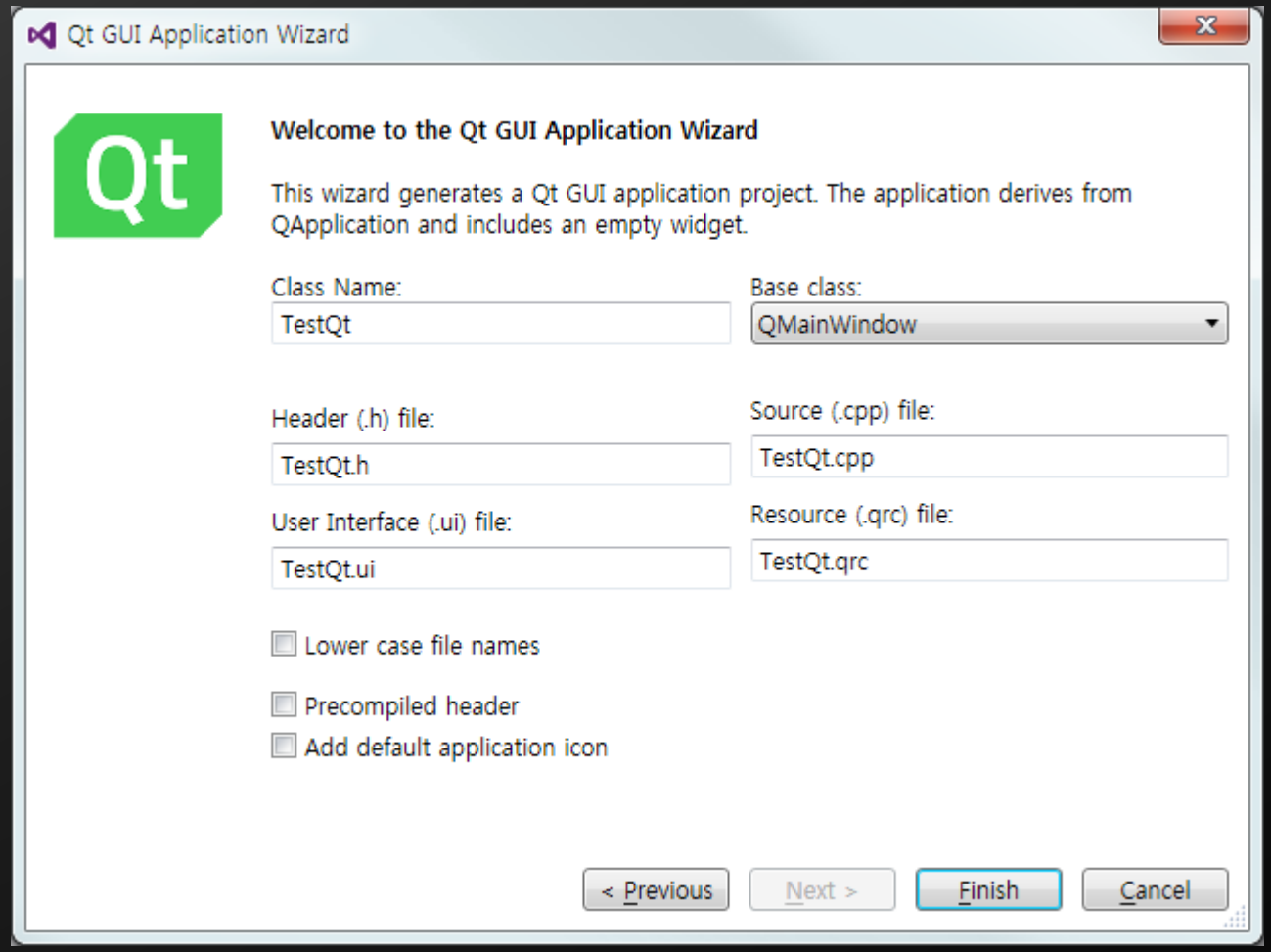
# Qt application

❖ Creating a Qt application



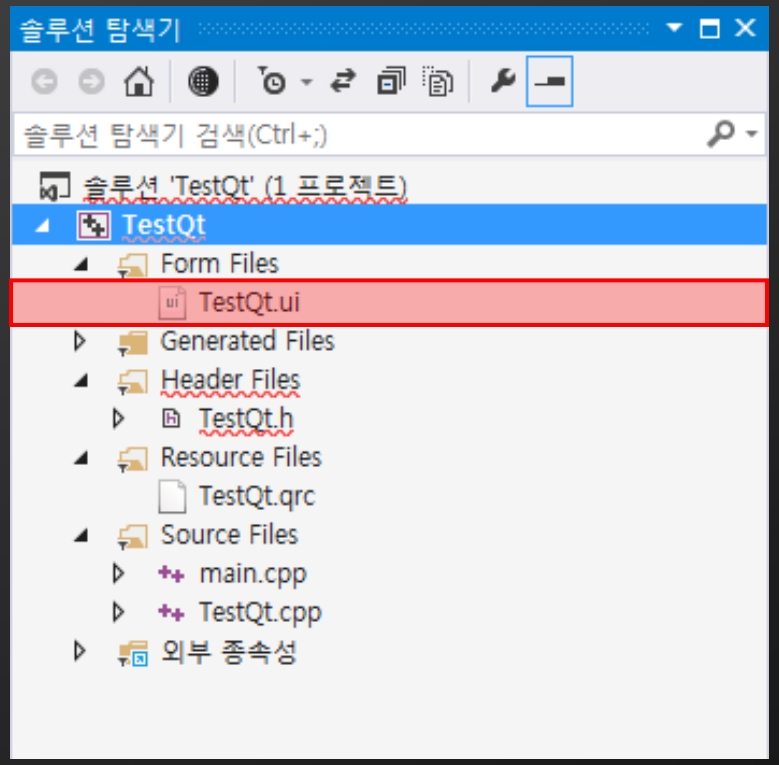
# Qt application

- Creating a Qt application



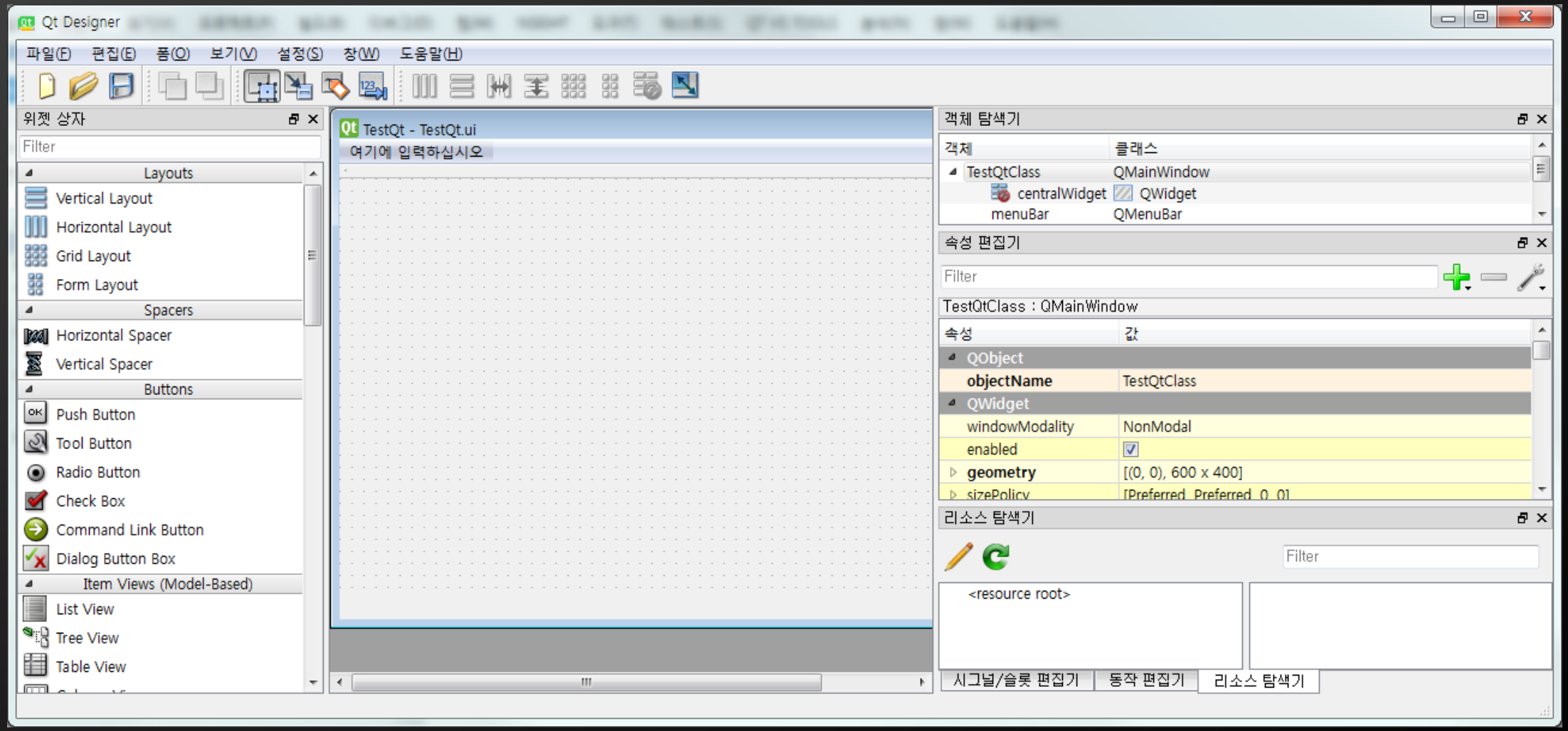
# Qt application

❖ **Structure of Qt application**



# Qt application

## ❖ Structure of Qt application



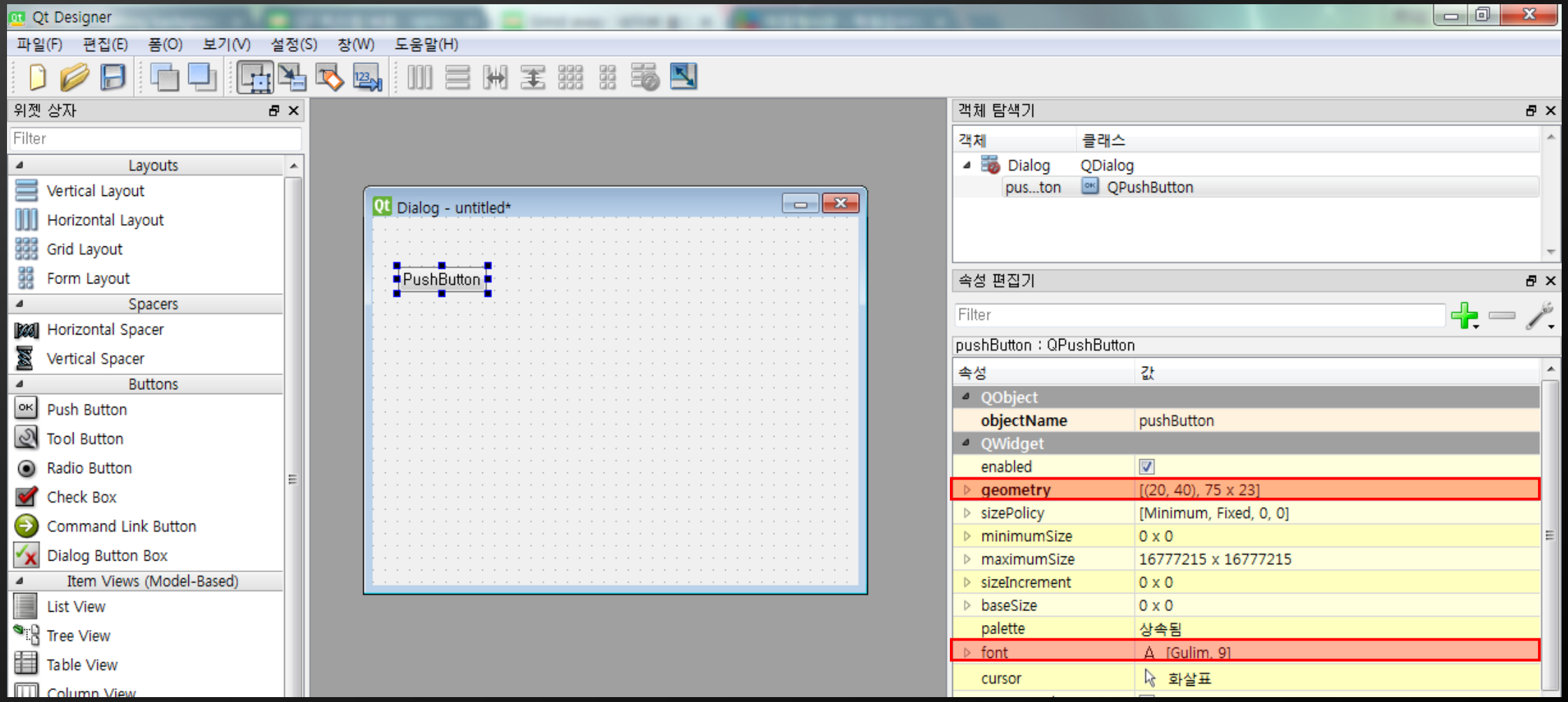
# Qt application

- **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) |

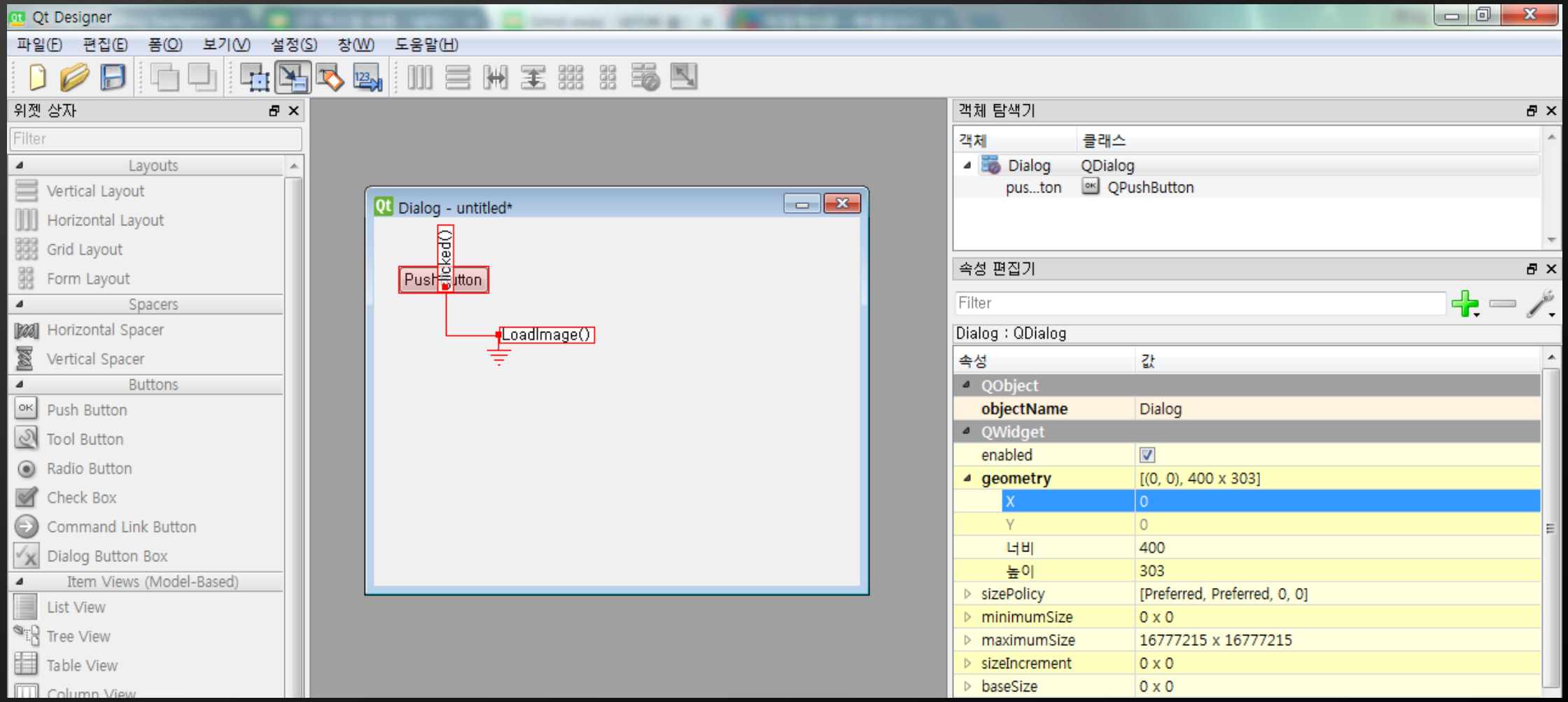
# Qt application

## ❖ Concept of signal and slot



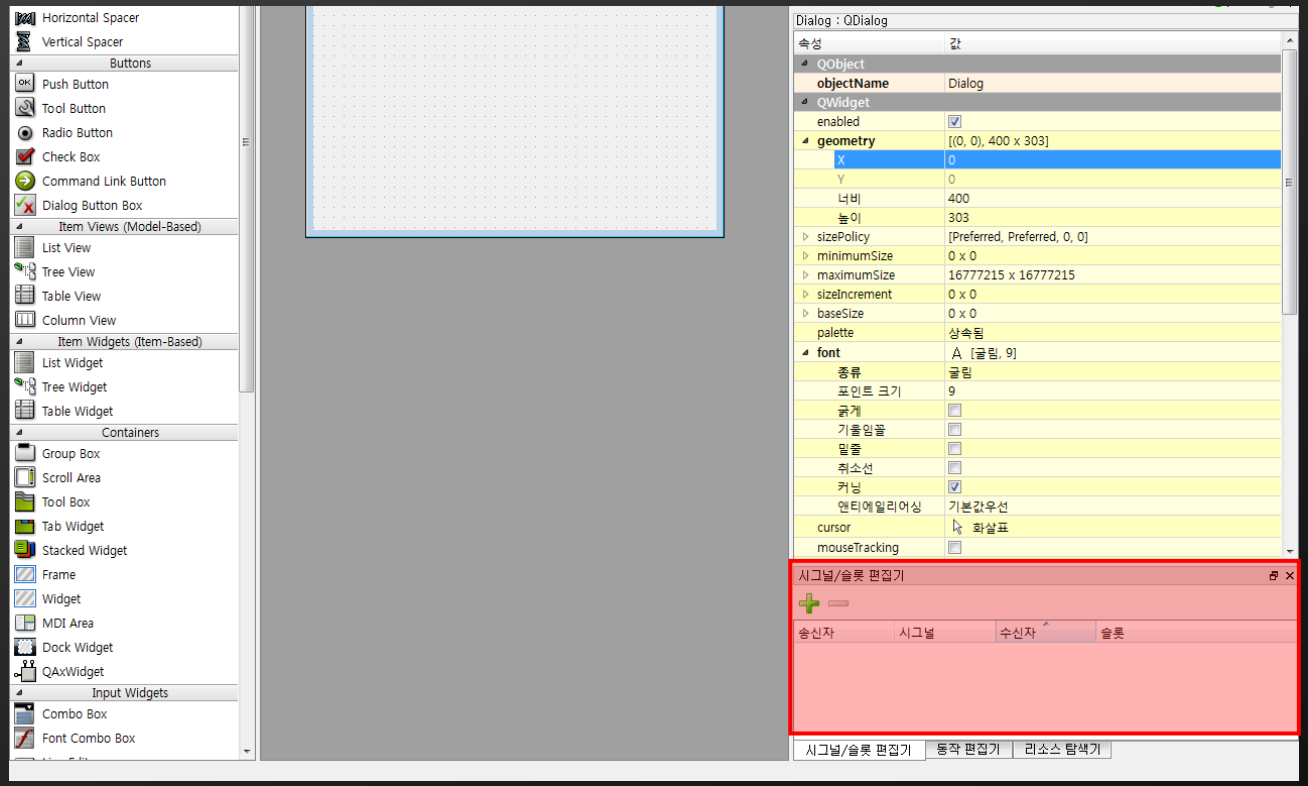
# Qt application

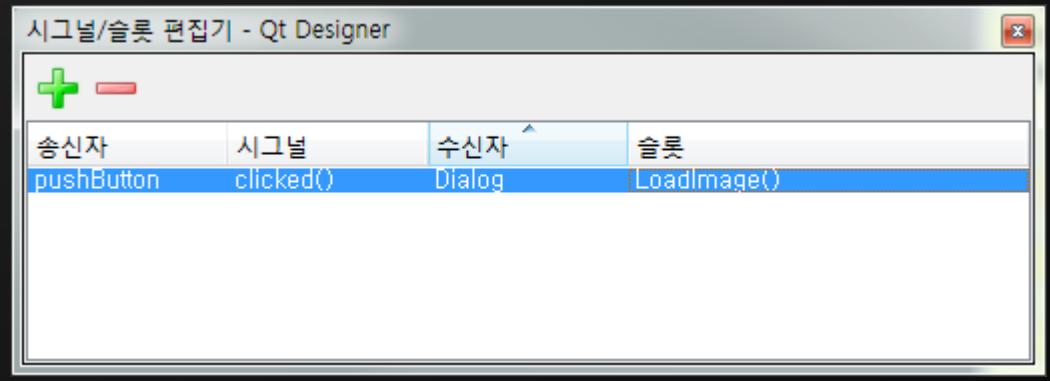
## ❖ Concept of signal and slot



# Qt application

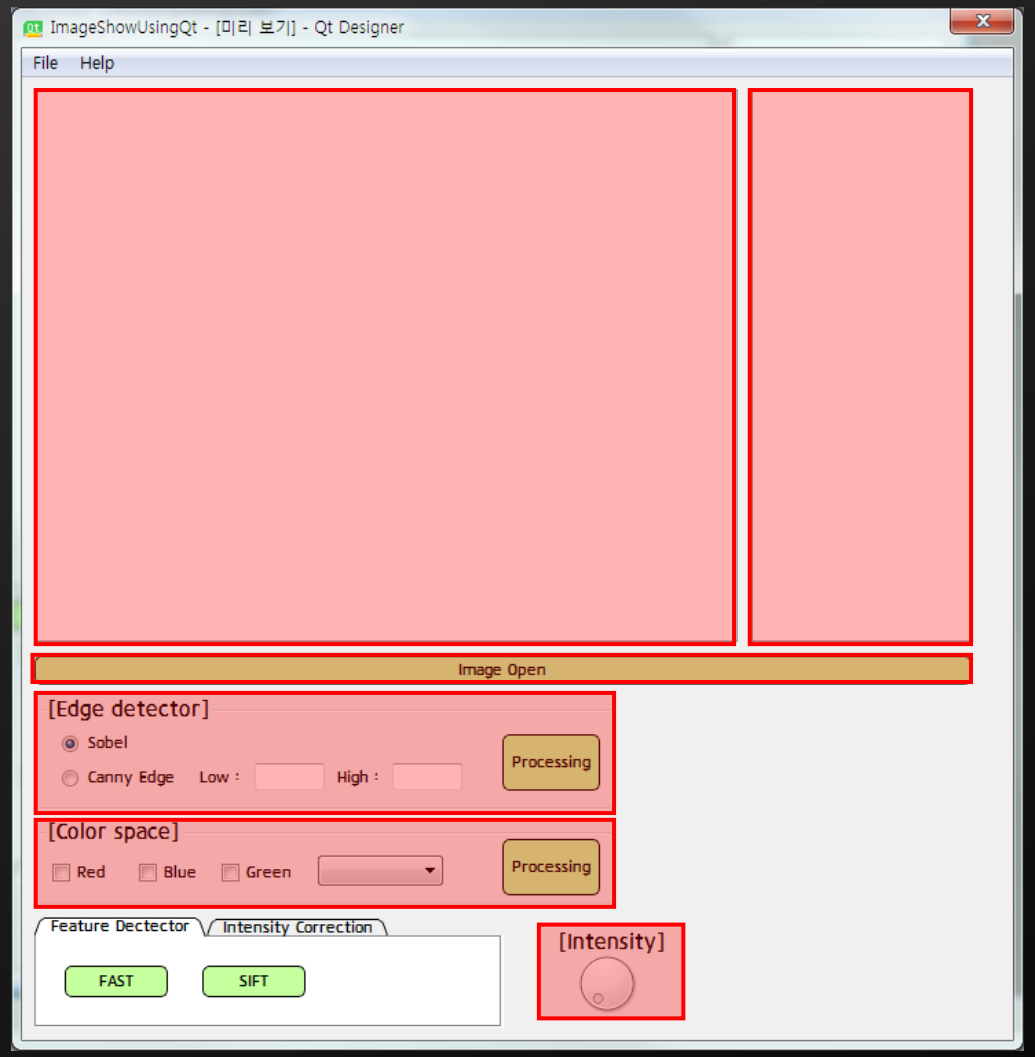
## ❖ Concept of signal and slot



# 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

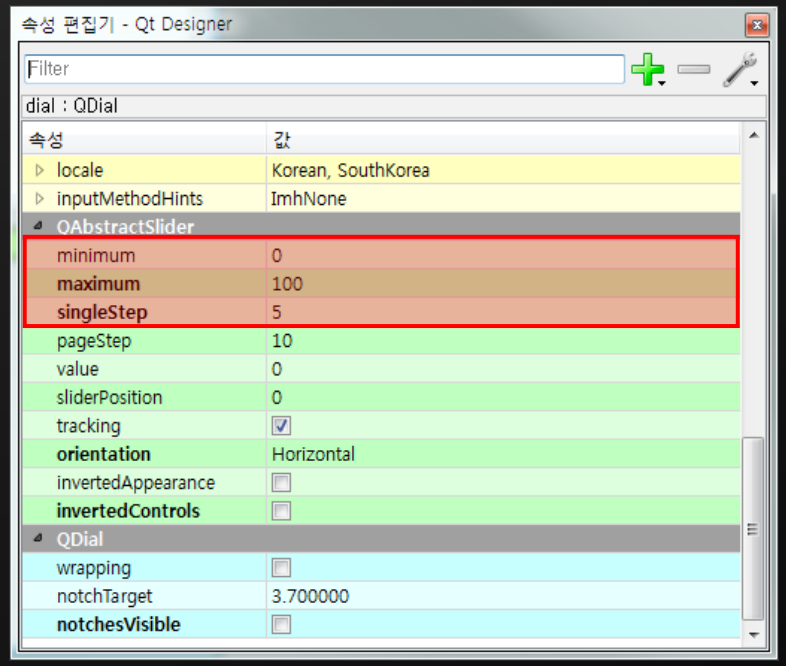
```
void ImageShowUsingQt::Activation()
{
	int i = ui.comboBox->currentIndex();
	if (i == 0)
	{
		ui.RedButton->setCheckable(1);
		ui.BlueButton->setCheckable(1);
		ui.GreenButton->setCheckable(1);
	}

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

# Qt application

❖ Dial

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



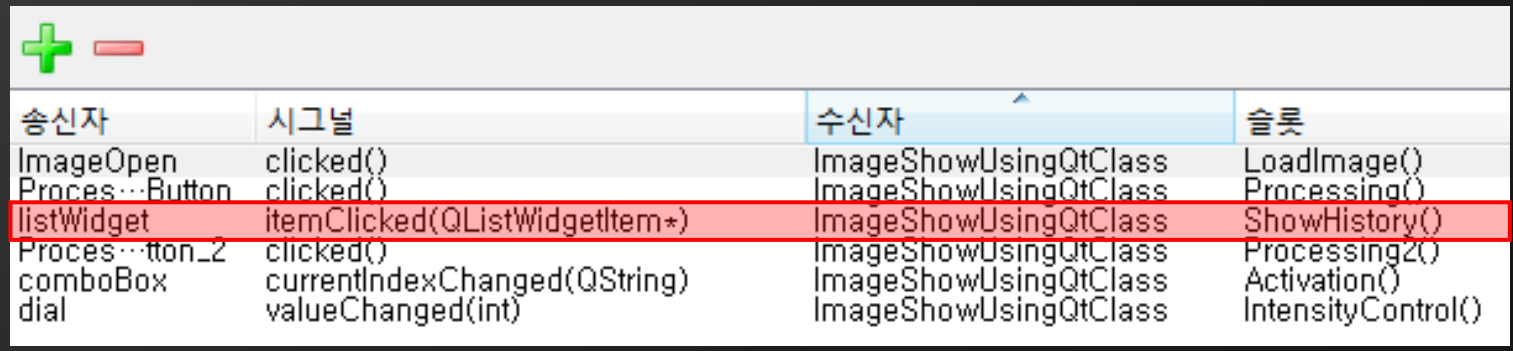
# Qt application

## ❖ 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() |
| Proces…Button | clicked() | ImageShowUsingQtClass | Processing() |
| listWidget | itemClicked(QListWidgetItem*) | ImageShowUsingQtClass | ShowHistory() |
| Proces…tton_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

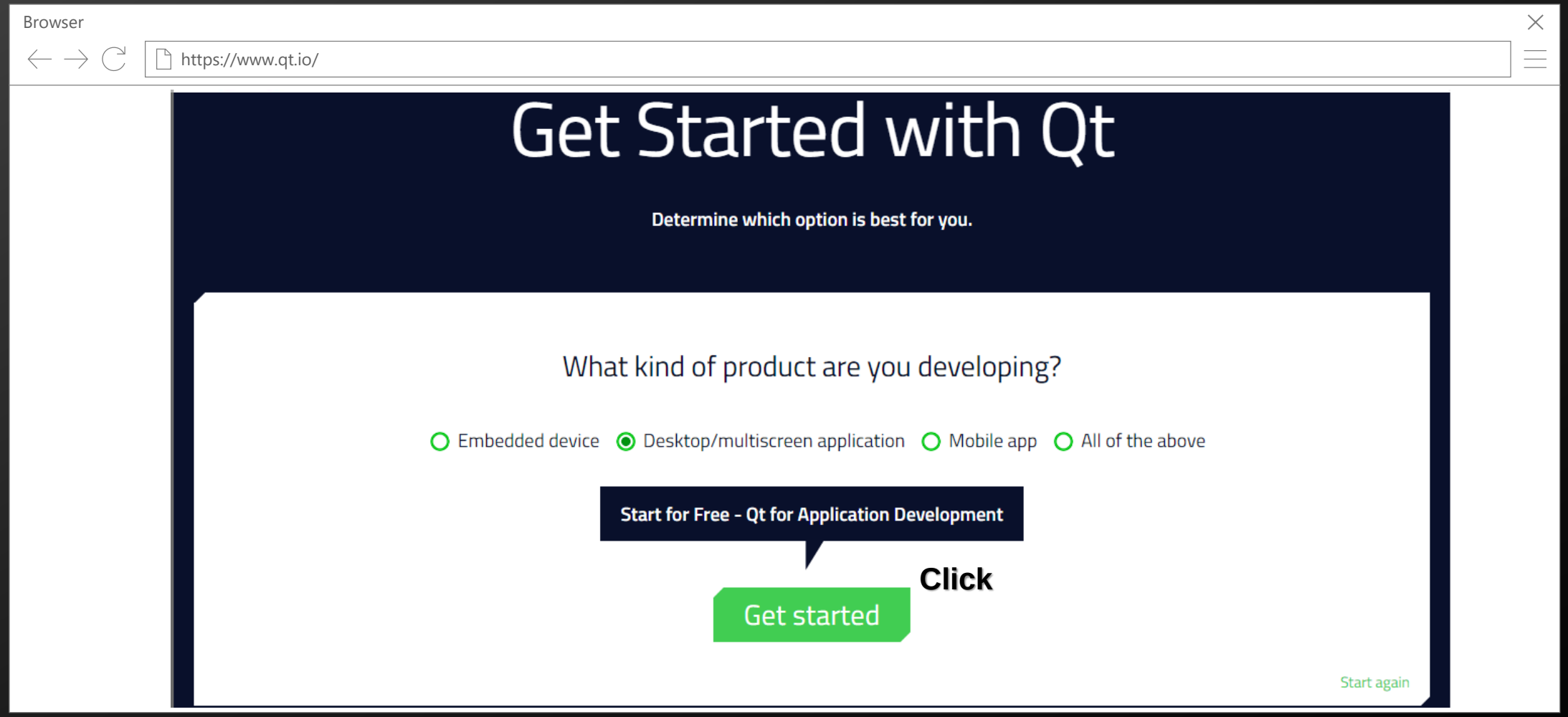


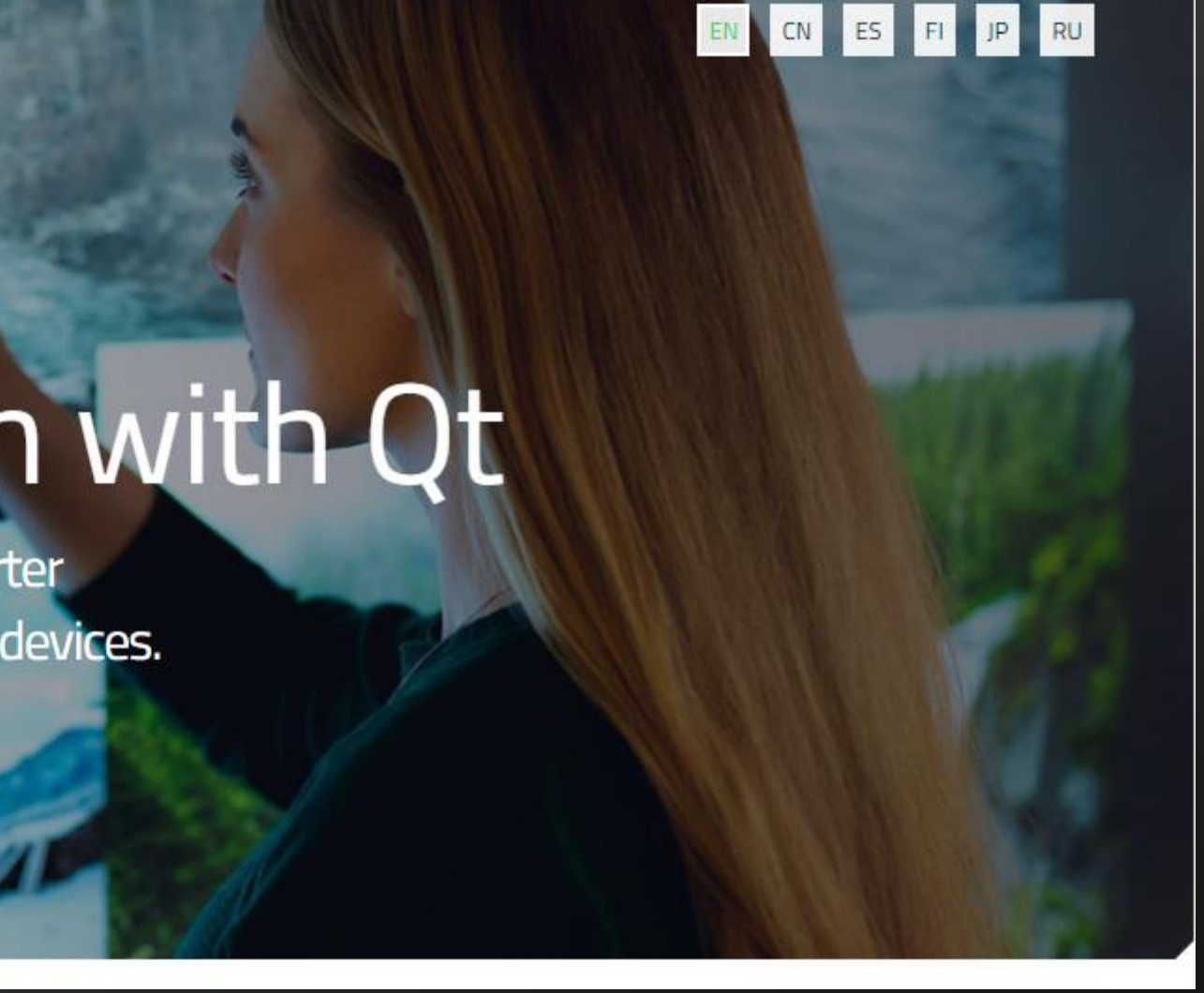
# Q & A

Thank You!!!

# Installation

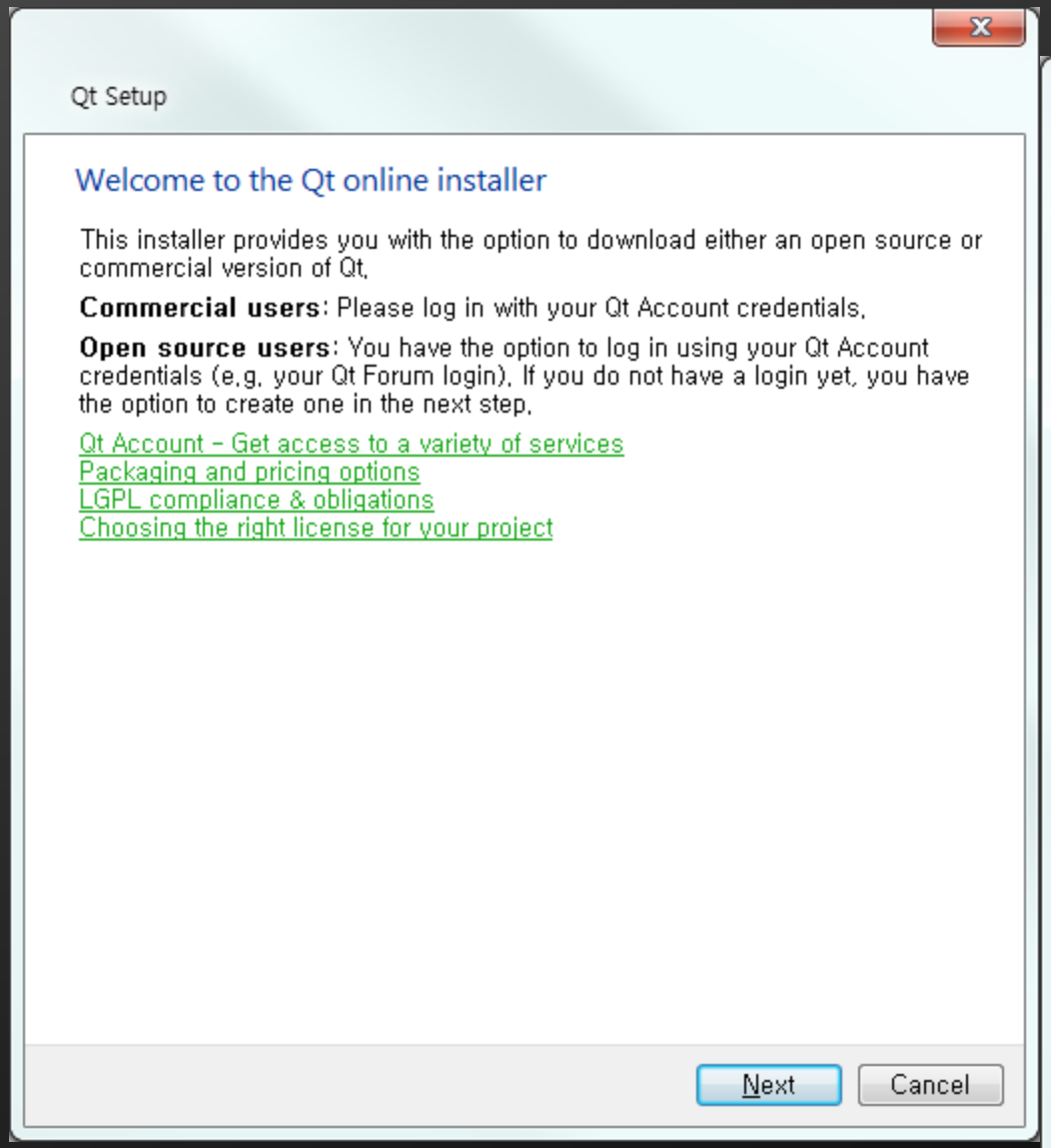
- Download Qt

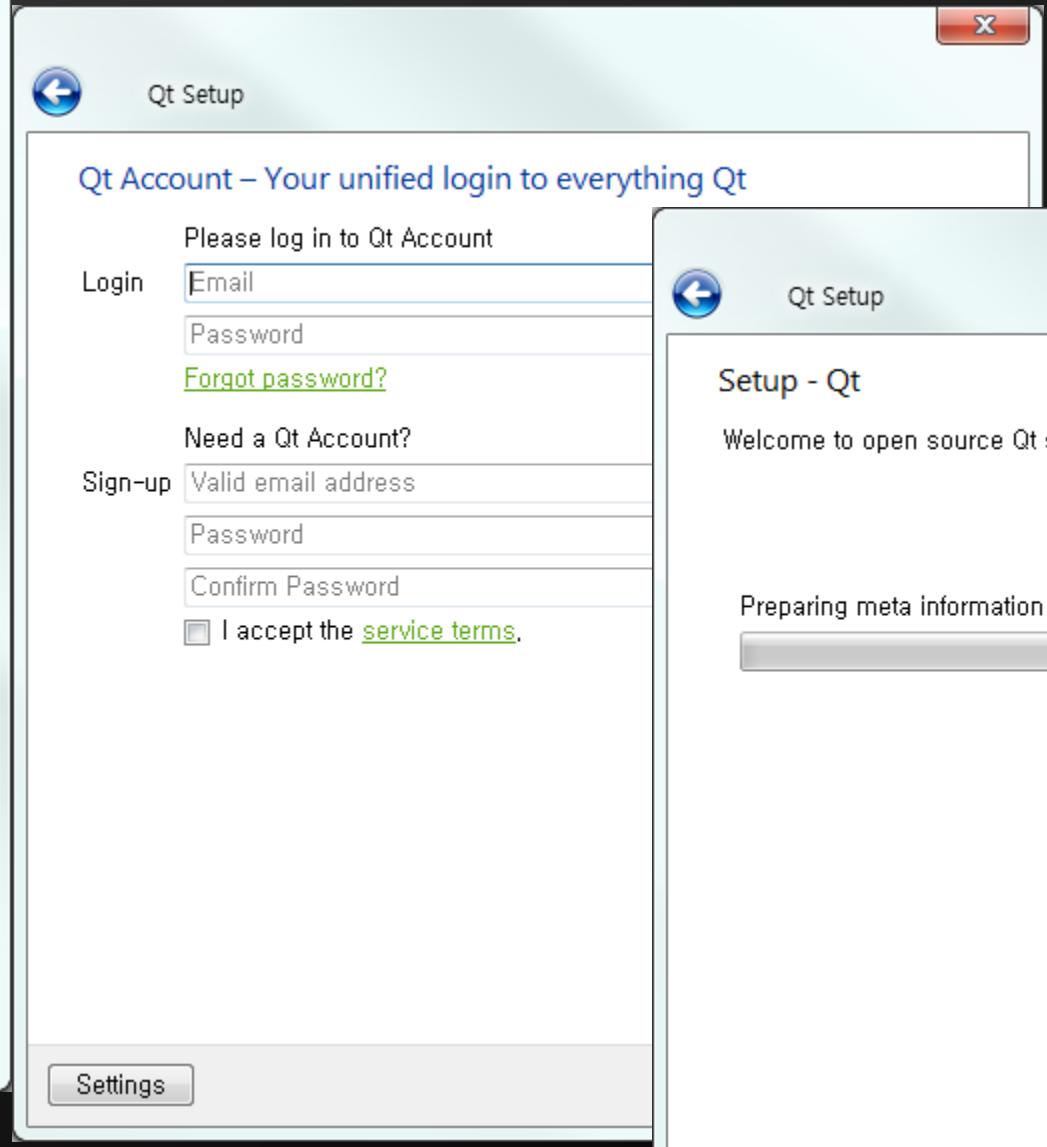


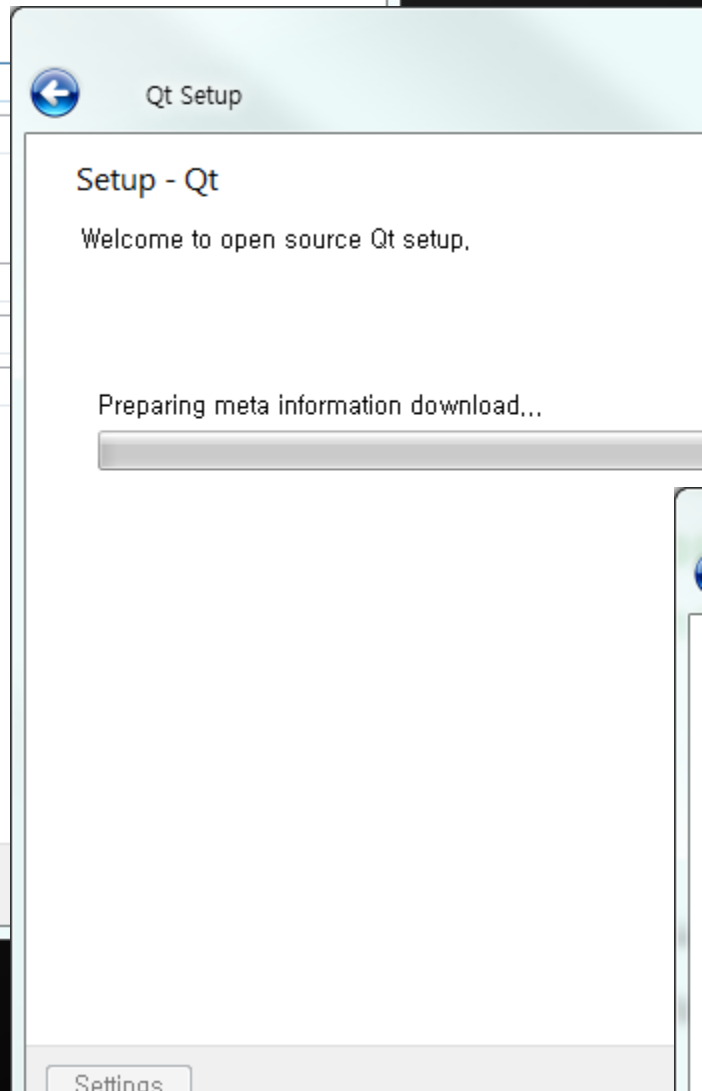



There a

Comm


Qt Setup
Welcome to the Qt online installer
This installer provides you with the option to download either an open source or commercial version of Qt.
Commercial users: Please log in with your Qt Account credentials.
Open source users: You have the option to log in using your Qt Account credentials (e.g. your Qt Forum login). If you do not have a login yet, you have the option to create one in the next step.
Qt Account - Get access to a variety of services
Packaging and pricing options
LGPL compliance & obligations
Choosing the right license for your project
Next
Cancel


Qt Setup
Qt Account – Your unified login to everything Qt
Please log in to Qt Account
Login
Email
Password
Forgot password?
Need a Qt Account?
Sign-up
Valid email address
Password
Confirm Password
I accept the service terms.
Settings


Qt Setup
Setup - Qt
Welcome to open source Qt setup.
Preparing meta information download...
Settings

확장 및 업데이트

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

정렬 기준: 관련성

qt

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

**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+...

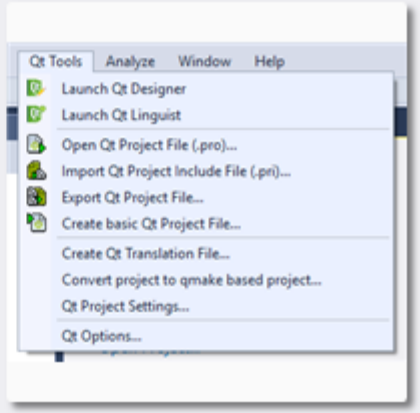
**BlackBerry Native Plugin for Visual Studio 2013**
Package providing Cascades and Native SDK support inside Visual Studio for PlayBook and BlackBerry 10 devices

**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




닫기(C)

Qt Options

Qt Versions | Qt Def

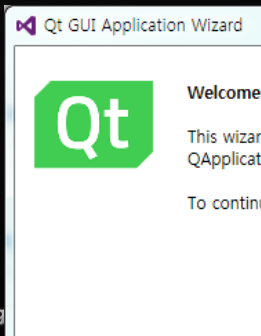
Name

Default Qt/Win ve

Qt GUI Application Wizard

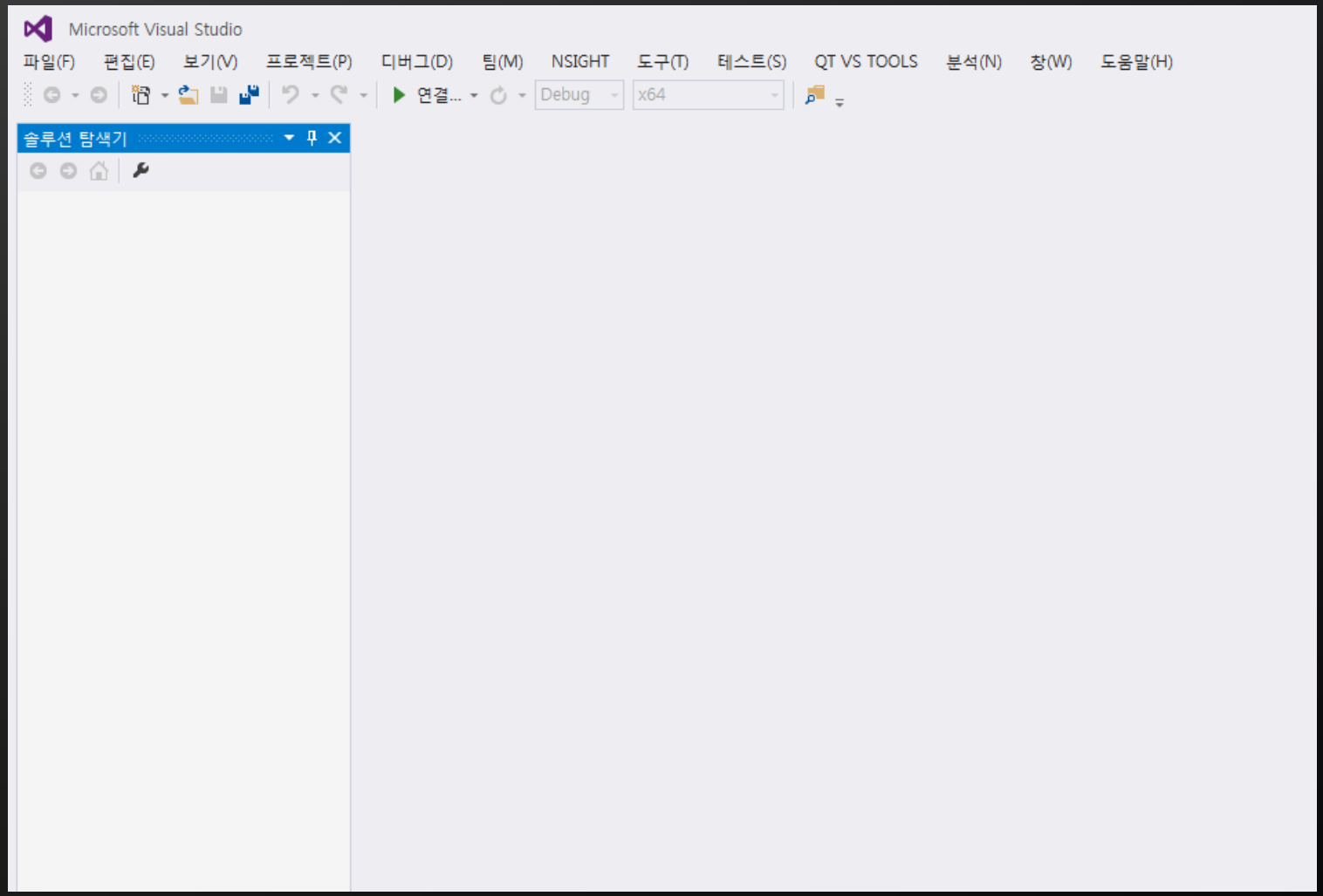
# Installation

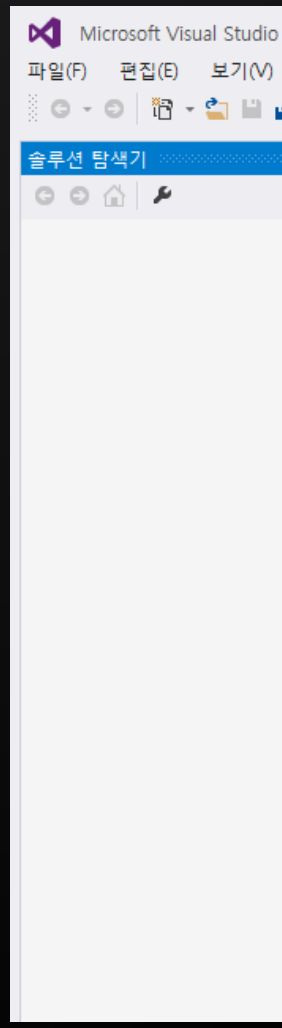
- Visual studio extension



# Installation

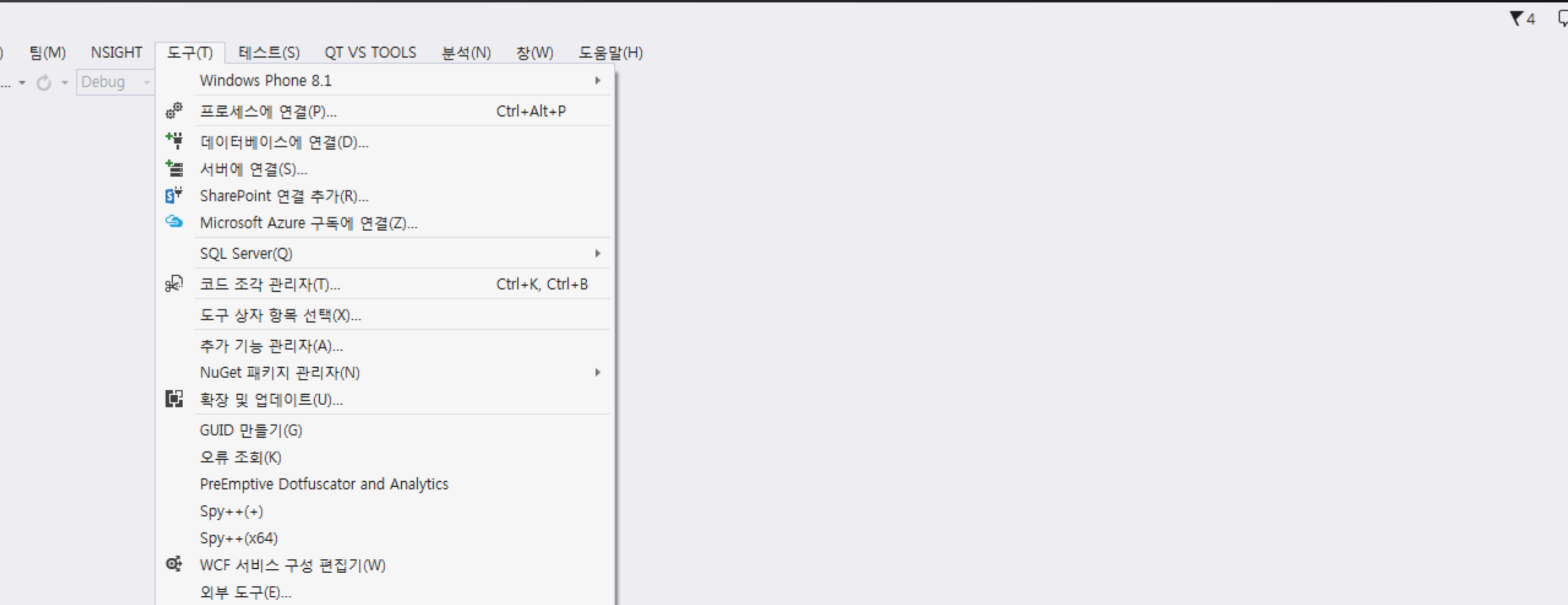
- Visual studio extension



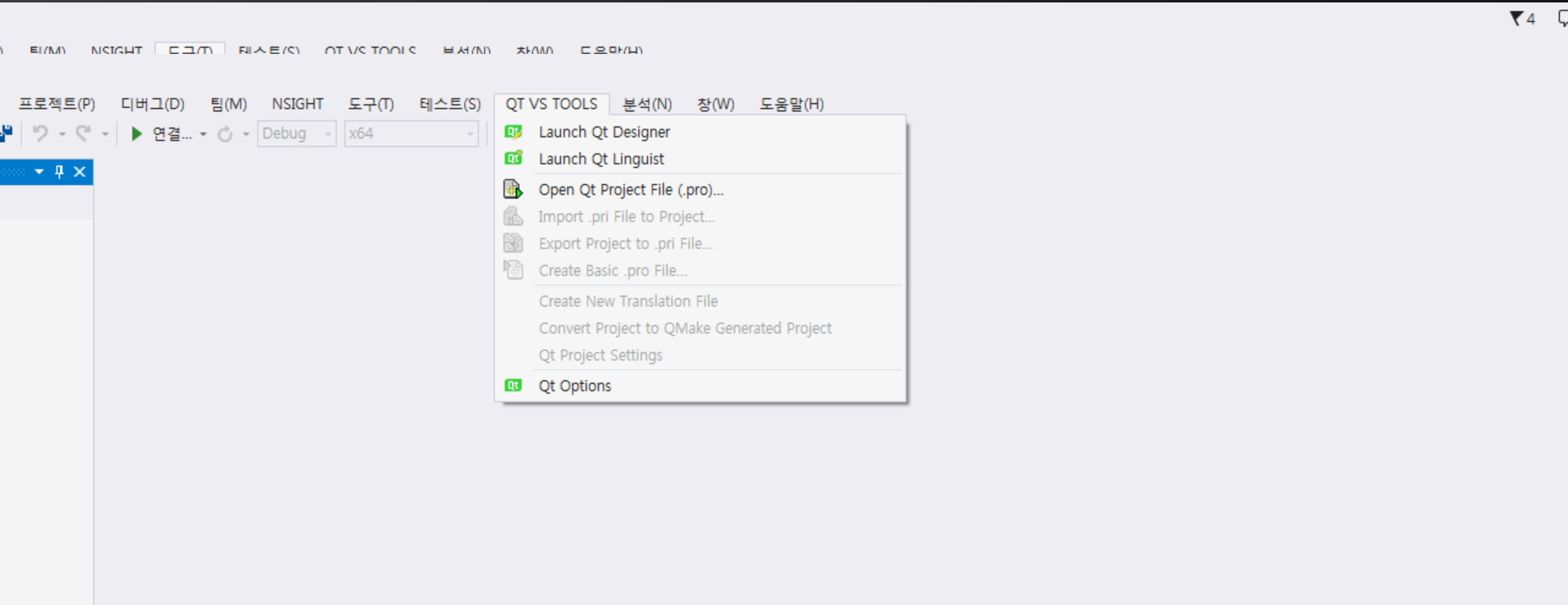
# Installation

❖ 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();
};
```