



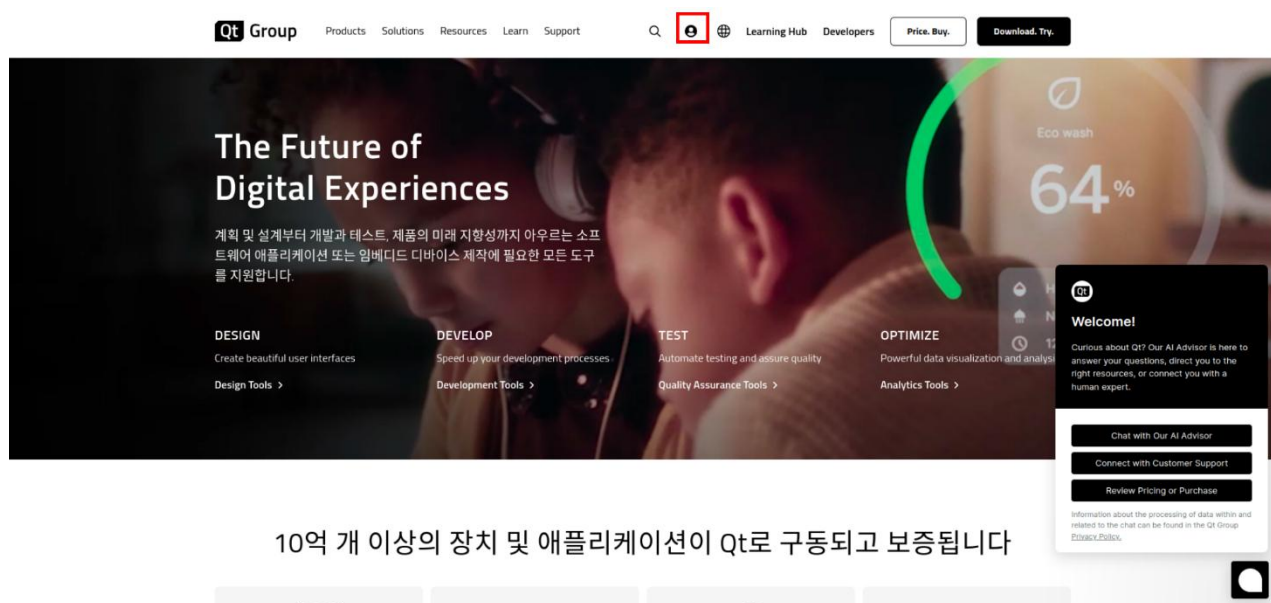
RO:BIT
ROBOT SPORT GAME TEAM

Team. RO:BIT | **QT**

18기 정유정

Qt 설치

-회원가입 진행



Qt Account

Your unified login to everything Qt

Sign in

[Forgot password?](#)


[Create Qt Account](#)

With Qt Account you get access to exclusive services, support and information. Use your Qt Account credentials to sign-in to all Qt services, including Qt Bugreports, Codereview, Forums & Wiki, and The Qt Company Support Center.

<https://www.qt.io/ko-kr/>

Qt 설치

-Open Source 라이선스 발급받아 사용



HomeMy profileSupport centerLog out

Your unified login to everything Qt

As a verified Qt Account user, you have exclusive access to the Qt Online Installer and [The Qt Company Support FAQ knowledge base](#).

Want to download Qt for open source usage?

Start here <https://www.qt.io/download-open-source>

1. Verify your Qt Account, if you haven't already
2. Understand your obligations as an open source user
3. Get the online installer and find the right binary for your development environment

Want a Qt commercial license? [Buy Qt Now](#).

Want to try before you buy? The Qt Company provides businesses with commercially viable development projects, a free Qt trial including all our commercial packages and components, plus access to the official Qt support desk for getting started assistance.

Get your free trial now

License and Access Information

Please note that if you initiate any Qt evaluation, whether for the Qt Framework and Tools or QA Tools, you will be redirected to the Customer Portal for further management and support. If your evaluation or commercial license expires, you will also be directed here for continued access and assistance.

Useful Quick Links:

- > [Documentation](#)
- > [Getting Started](#)
- > [Qt Blog](#)
- > [Qt Account](#)

Qt Group

ProductsSolutionsResourcesSupport

🔍👤🌐

Learning HubDevelopers

Price. Buy.

Download. Try.

Looking for Qt binaries?

Find them in the Qt Online Installer. It will steer you to the right download version and help you install tools and add-on components that are available for your open source license.

Download the Qt Online Installer

Frequently Asked Questions

> Why is Qt licensed also under an open source license?

> Why do you have an agreement with KDE about your licensing? What KDE is and what's the history of Qt and KDE?

> What is the consequence of not complying with the LGPL/GPL restrictions?

> Can I use the Community open source version to develop my commercial product?

| Qt 설치

- 사이트에서 설치

```
cd ~/Downloads  
sudo chmod +x qt_online_installer_x84_version.run  
./qt_online_installer_x84_version.run
```

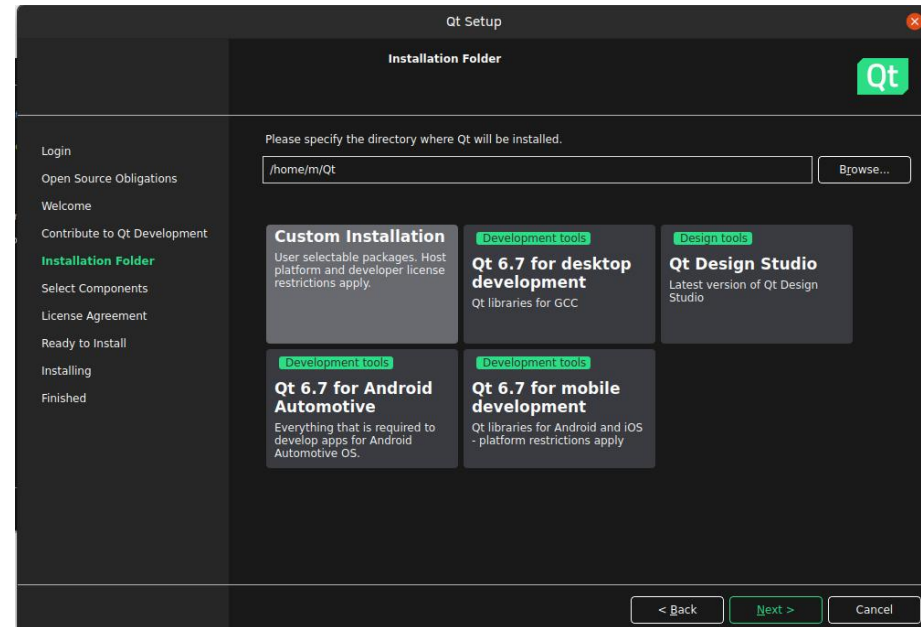
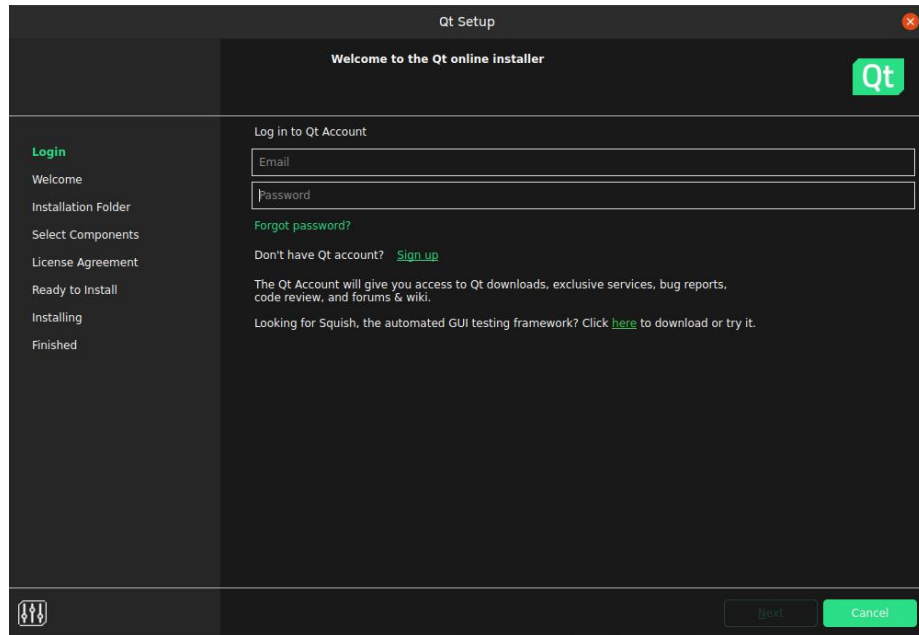
Or

- 터미널에서 설치

```
sudo apt install qtcreator
```

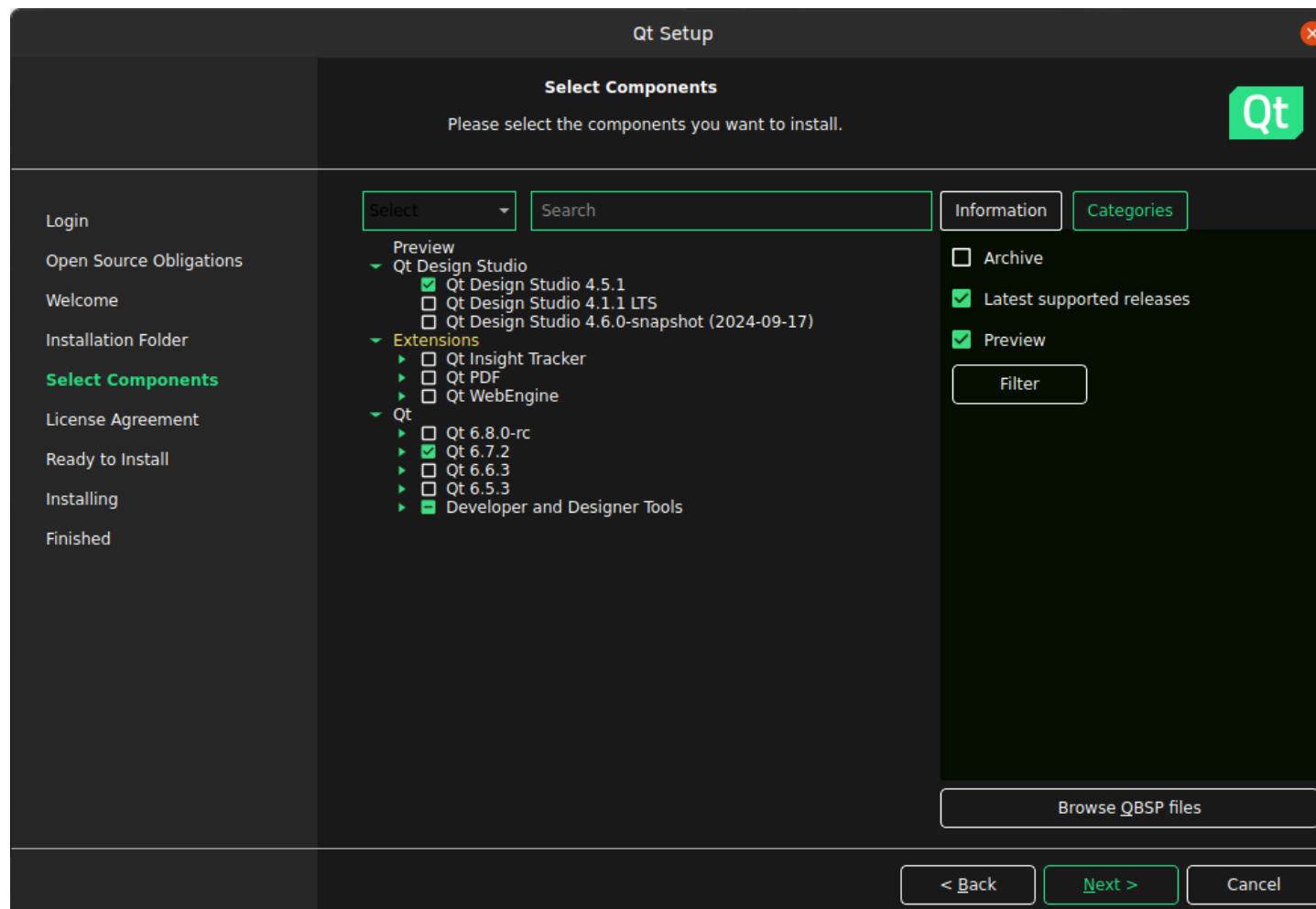
Qt 설치

- 로그인 진행



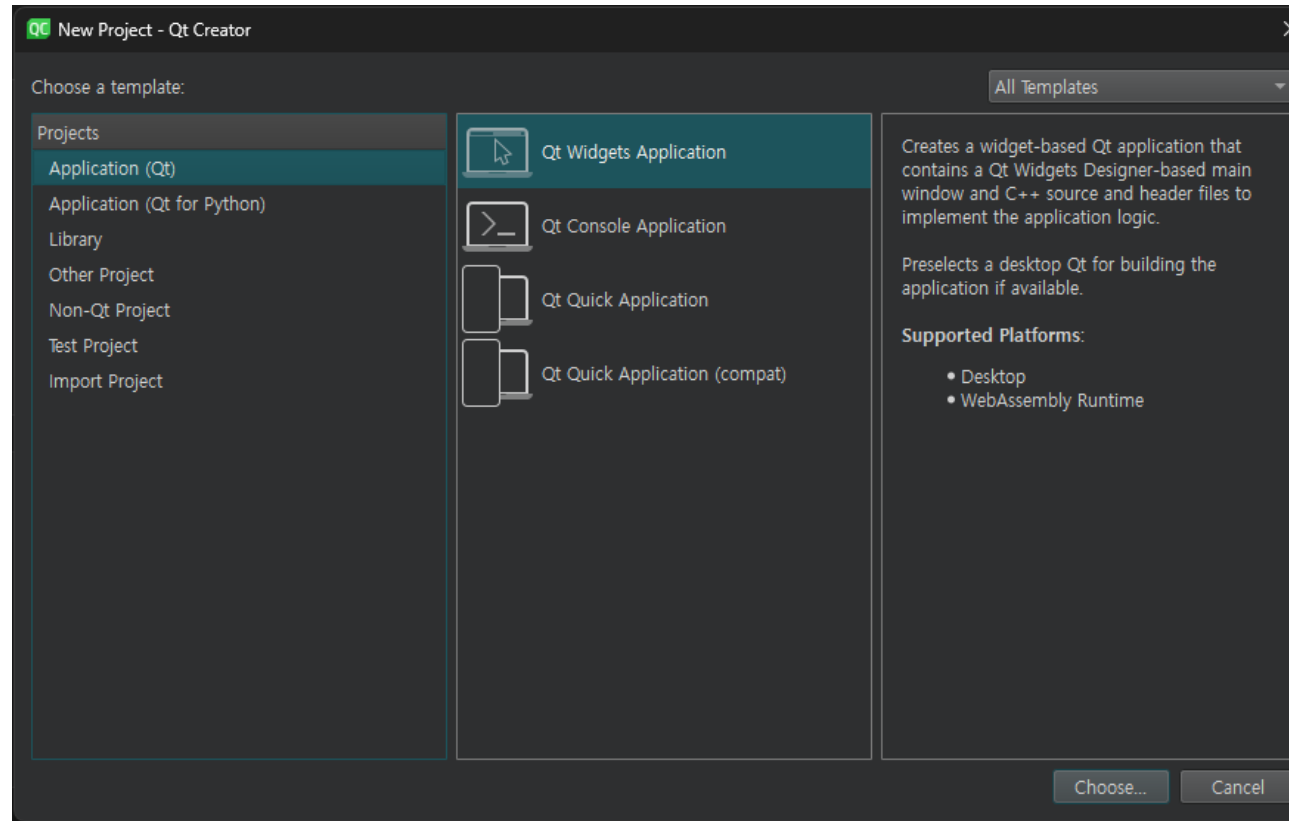
Qt 설치

- 설치 진행



| Qt 프로젝트 생성

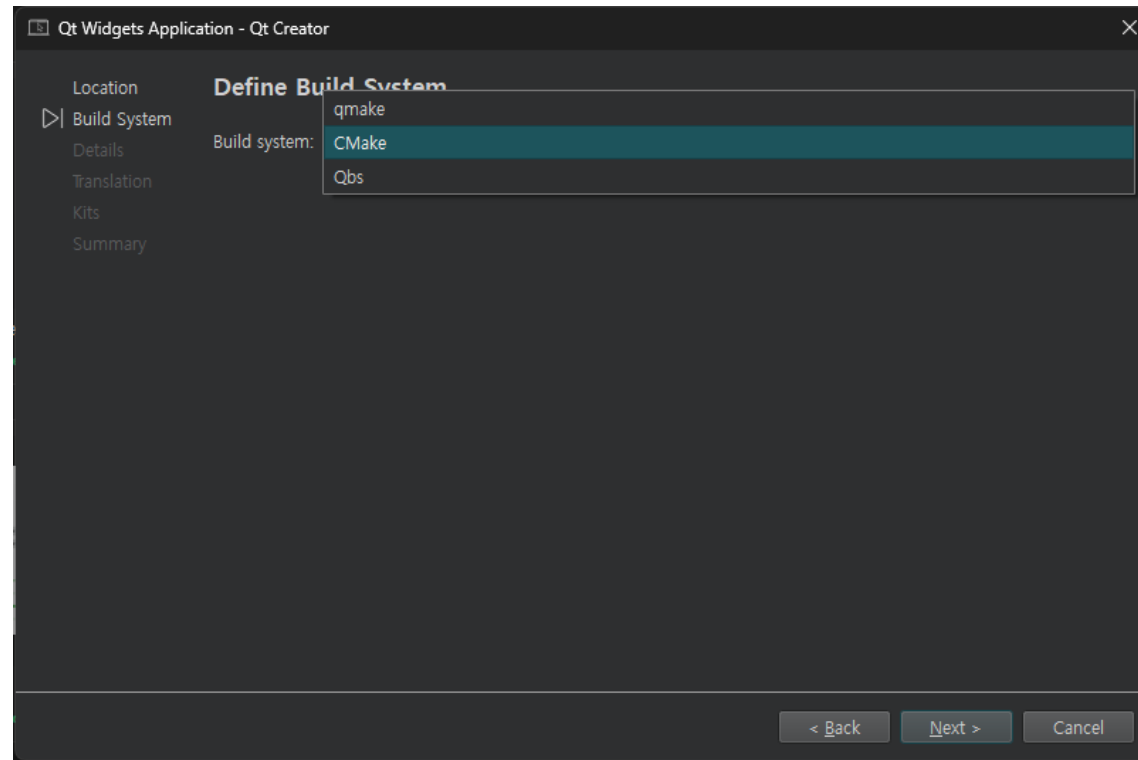
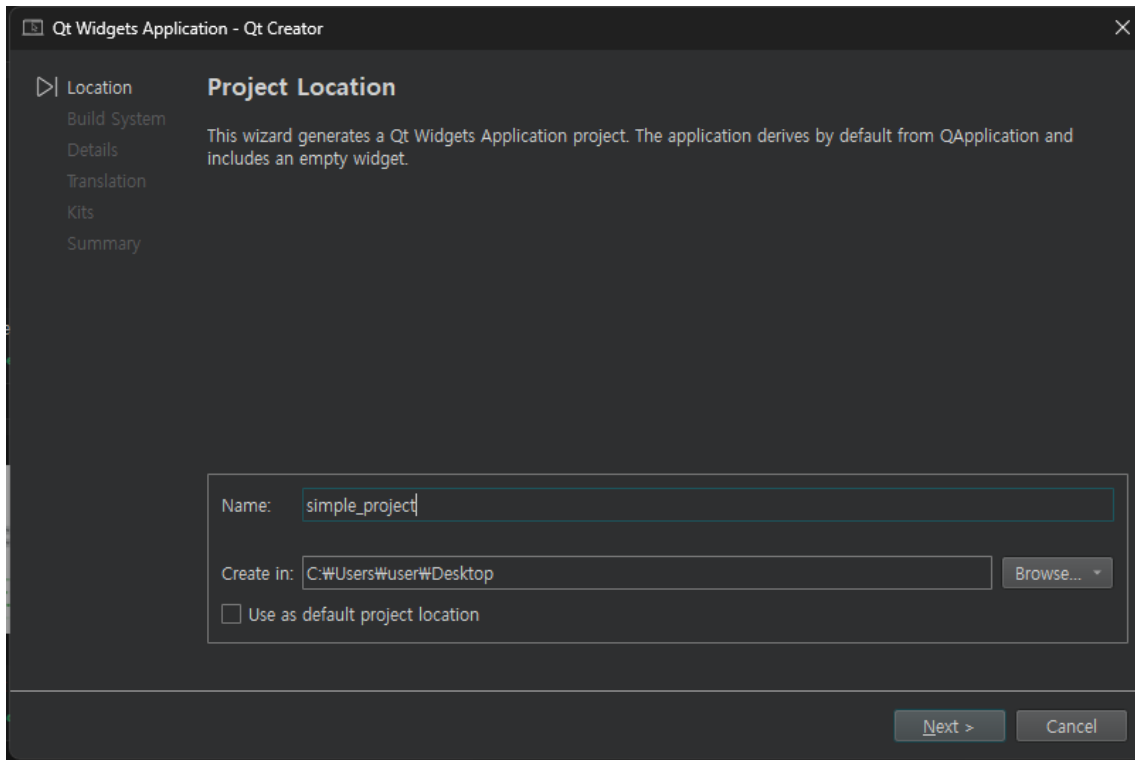
-UI



Qt Widgets Application으로 생성

Qt 프로젝트 생성

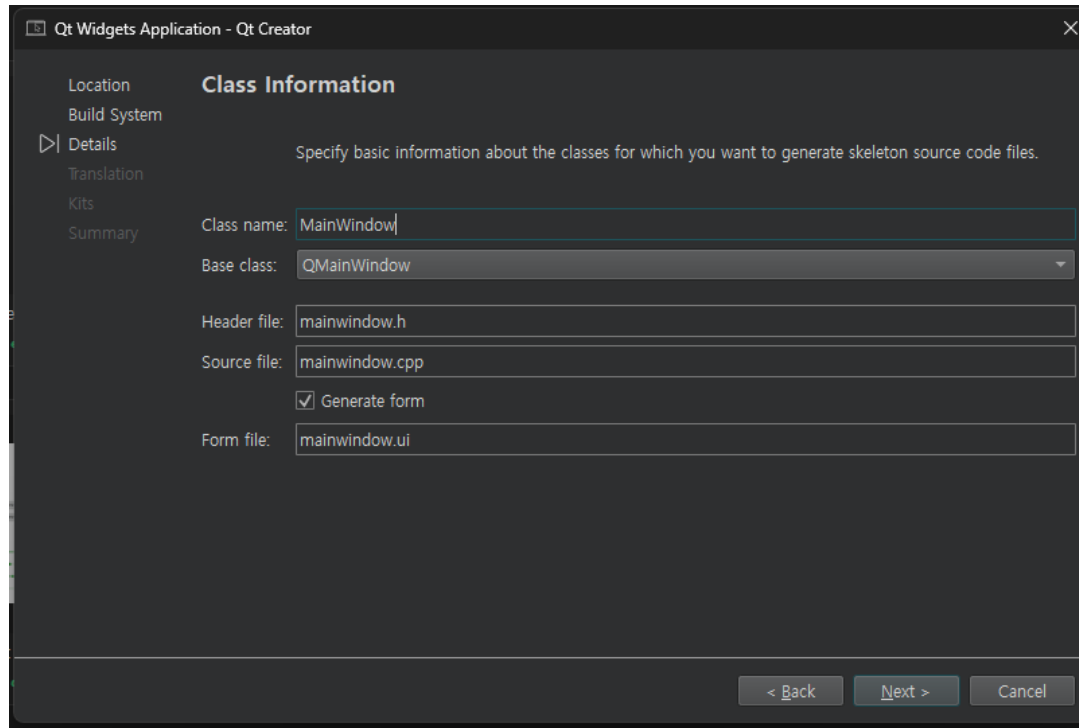
-UI



빌드 시스템은 CMake로 설정

Qt 프로젝트 생성

-UI



Qt Widgets Application - Qt Creator

Location
Build System
Details
Translation
Kits
Summary

Class Information

Specify basic information about the classes for which you want to generate skeleton source code files.

Class name:

Base class:

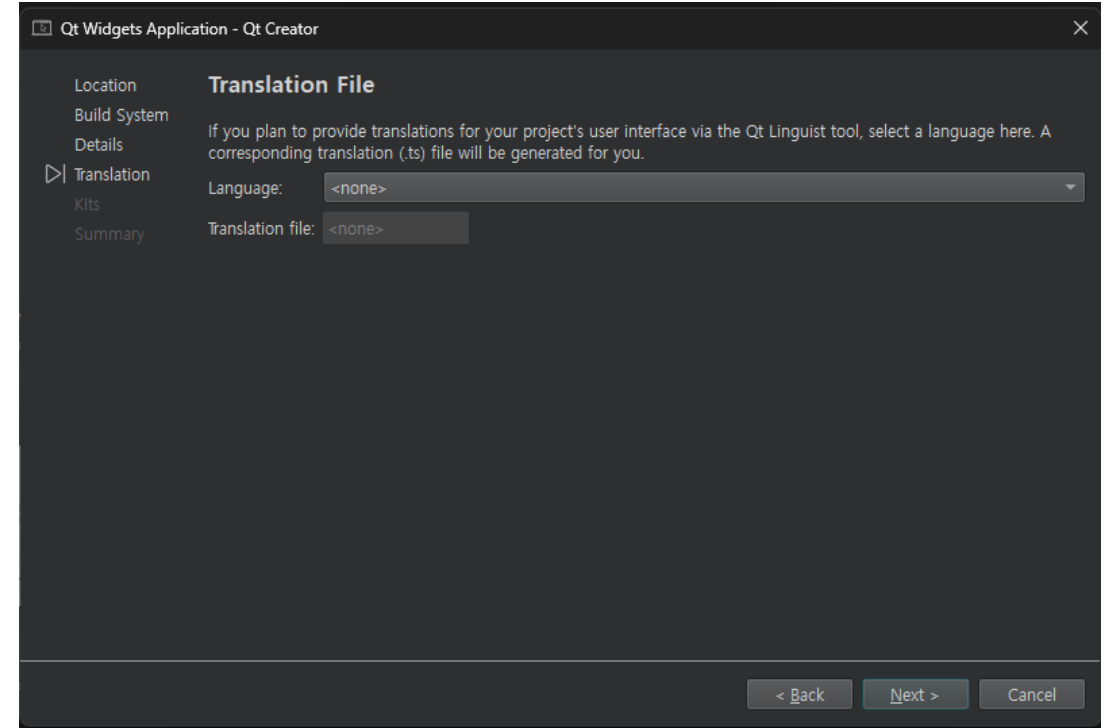
Header file:

Source file:

☒ Generate form

Form file:

< Back Next > Cancel



Qt Widgets Application - Qt Creator

Location
Build System
Details
Translation
Kits
Summary

Translation File

If you plan to provide translations for your project's user interface via the Qt Linguist tool, select a language here. A corresponding translation (.ts) file will be generated for you.

Language:

Translation file:

< Back Next > Cancel

헤더 및 소스 파일명 설정 가능

Qt 프로젝트 생성

-UI

```
Project_name
├── CMakeLists.txt
├── include
│   └── mainwindow.h
├── resources
│   ├── images
│   │   └── icon.png
│   └── resources.qrc
├── src
│   ├── main.cpp
│   └── mainwindow.cpp
└── ui
    └── mainwindow.ui
```

위와 같은 구조로 설정

CMakeLists

```
cmake_minimum_required(VERSION 3.5)
project(qt_template VERSION 1.0 LANGUAGES CXX)
```

```
# Set C++ standard
set(CMAKE_CXX_STANDARD 11)
set(CMAKE_CXX_STANDARD_REQUIRED ON)
```

```
find_package(Qt6 COMPONENTS Widgets QUIET)
if(Qt6_FOUND)
    set(QT_VERSION_MAJOR 6)
    message(STATUS "Found Qt6: ${Qt6_VERSION}")
else()
    find_package(Qt5 COMPONENTS Widgets REQUIRED)
    set(QT_VERSION_MAJOR 5)
    message(STATUS "Found Qt5: ${Qt5_VERSION}")
endif()

if(NOT Qt5_FOUND AND NOT Qt6_FOUND)
    message(FATAL_ERROR "Qt5 or Qt6 not found. Please install Qt.")
endif()
```

project명령어를 통해 프로젝트명 지정, 버전 및 언어 지정

Qt5, 6 모두 지원 가능하도록 설정

CMakeLists

- 소스 파일 및 헤더 파일 설정

```
# Source files
set(SOURCES
src/main.cpp
src/mainwindow.cpp
)

# Header files
set(HEADERS
include/mainwindow.h
)

# UI files
set(UI_FILES
ui/mainwindow.ui
)

# Resource file
set(RESOURCE_FILES
resources/resources.qrc
)
```

- MOC, UI 및 리소스 파일 처리

```
if(QT_VERSION_MAJOR EQUAL 6)
    qt6_wrap_cpp(MOC_SOURCES ${HEADERS})
    qt6_wrap_ui(UIC_SOURCES ${UI_FILES})
    qt_add_resources(RESOURCE_SOURCES ${RESOURCE_FILES})
else()
    qt5_wrap_cpp(MOC_SOURCES ${HEADERS})
    qt5_wrap_ui(UIC_SOURCES ${UI_FILES})
    qt5_add_resources(RESOURCE_SOURCES ${RESOURCE_FILES})
endif()
```

〈- 리소스가 있을 시 설정 (UI에 이미지 등 아이콘 첨부 시 사용)

CMakeLists

- 링크 설정

```
add_executable(${PROJECT_NAME} ${SOURCES} ${MOC_SOURCES} ${UIC_SOURCES} ${RESOURCE_SOURCES})
```

```
# Link the appropriate Qt Widgets library
```

```
if(QT_VERSION_MAJOR EQUAL 6)
```

```
    target_link_libraries(${PROJECT_NAME} Qt6::Widgets)
```

```
else()
```

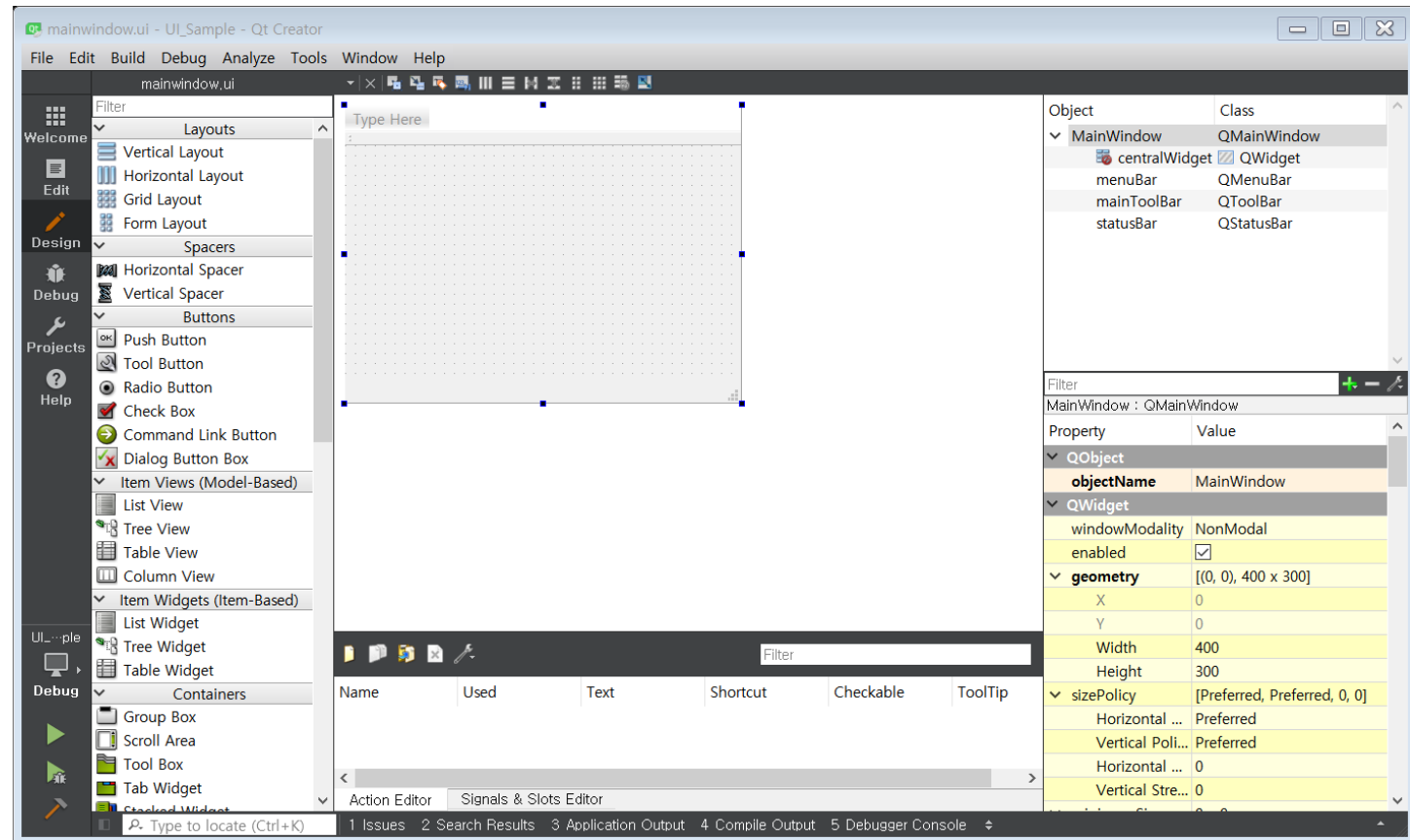
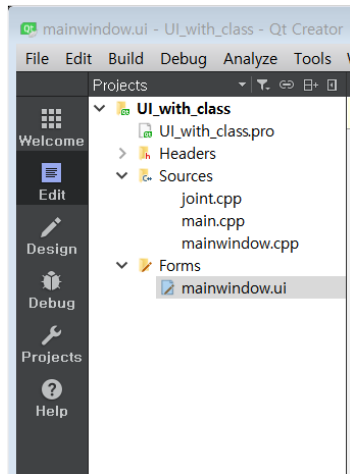
```
    target_link_libraries(${PROJECT_NAME} Qt5::Widgets)
```

```
endif()
```

qt

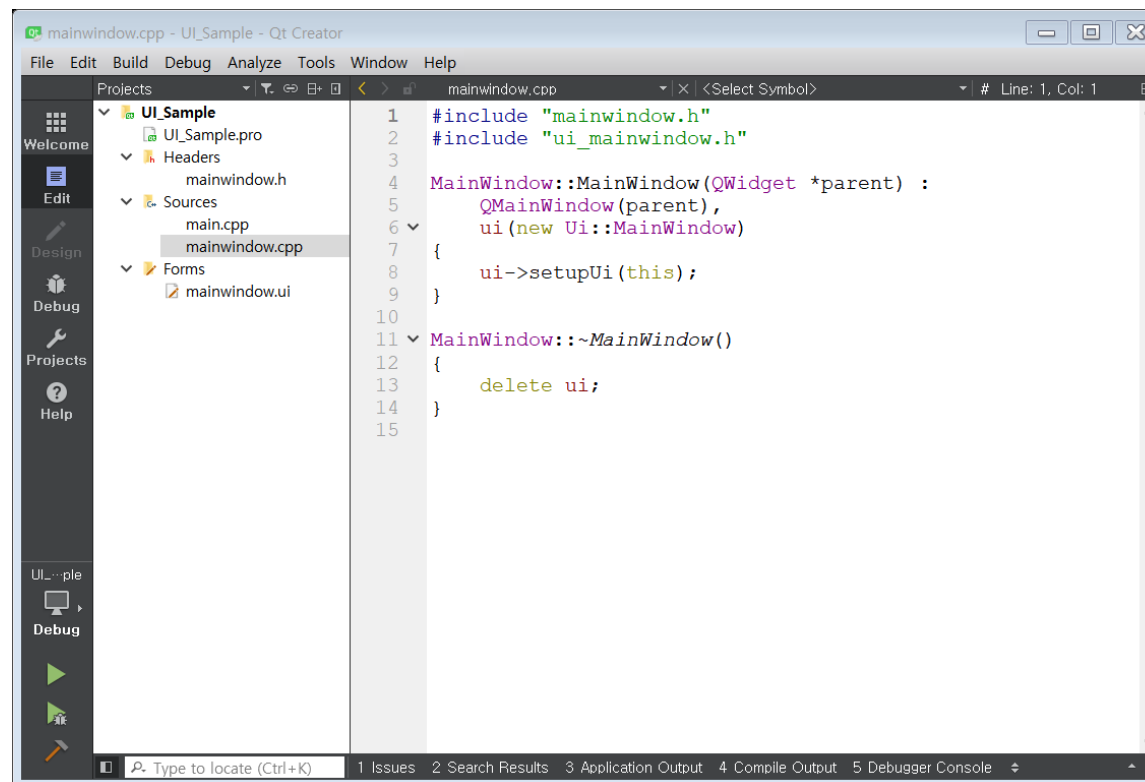
-UI

.ui파일을 사용해 UI 구성



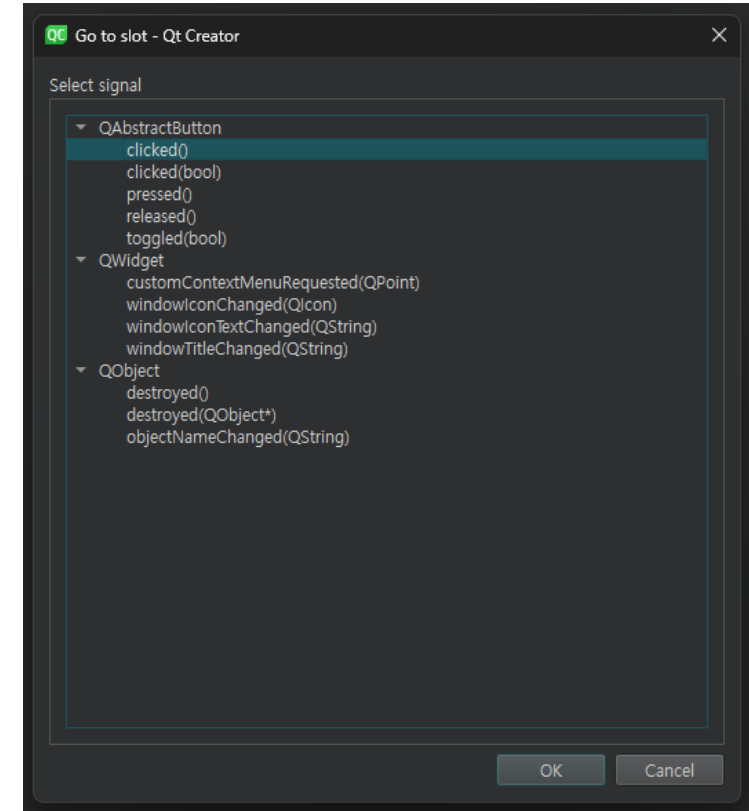
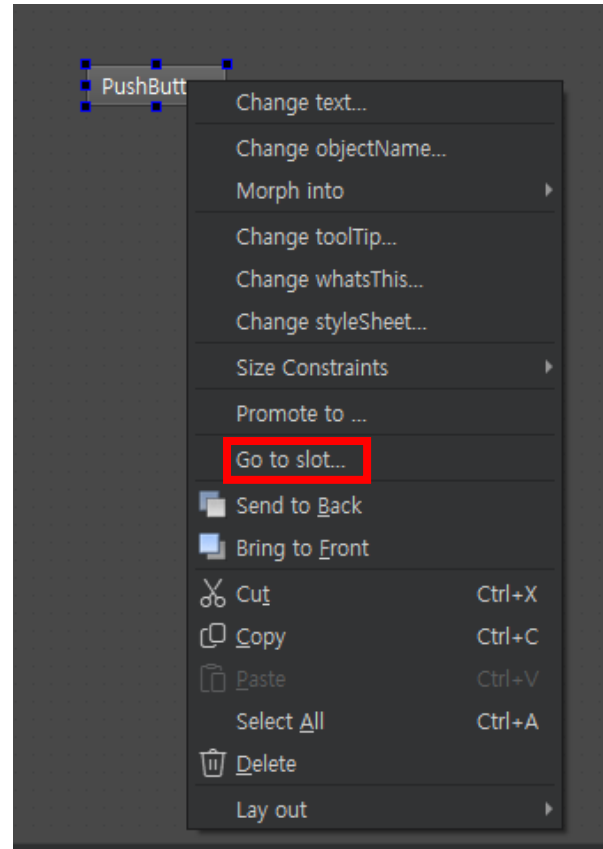
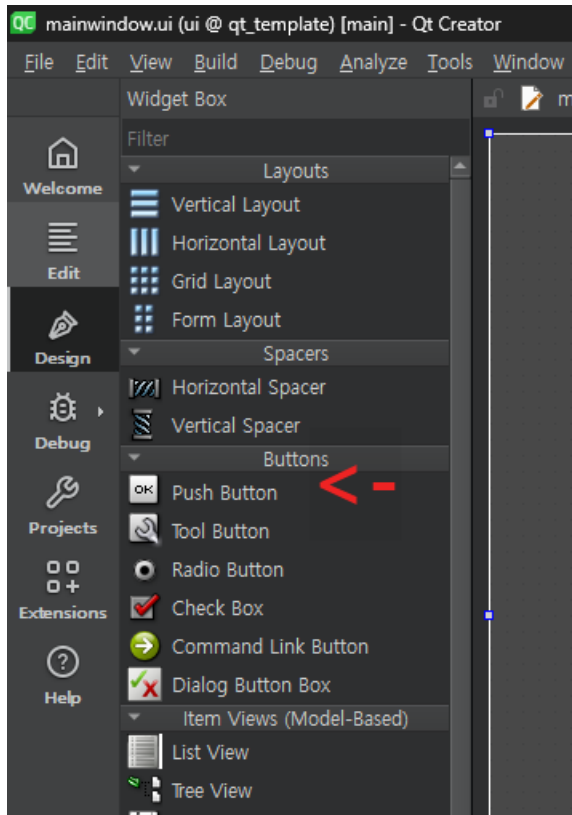
-UI

주로 mainwindow 클래스에서 코드 작업 진행



-UI

소스 및 헤더 파일에 signal, slot 등록



-UI

on_〈ui 이름〉_〈시그널 이름〉 () 과 같은 형태로 작성

```
#include "../ui_mainwindow.h"
#include "mainwindow.h"

MainWindow::MainWindow(QWidget* parent) : QMainWindow(parent)
{
    ui->setupUi(this);
    QIcon icon(":/image/images/icon.png");
    setWindowIcon(icon);
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_start_btn_clicked()
{
}
```

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <iostream>

QT_BEGIN_NAMESPACE
namespace Ui
{
    class MainWindow;
}
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget* parent = nullptr);
    ~MainWindow();

private slots:
    void on_start_btn_clicked();

private:
    Ui::MainWindow* ui;
};

#endif // MAINWINDOW_H
```

signal, slot, connect

connect(SIGNAL이 발생하는 곳 , 발생SIGNAL , SLOT이 발생하는 곳, 발생SLOT)

SIGNAL: 신호. 인자 전달도 가능.

SLOT:신호 발생 시 실행되는 함수

Connect : SIGNAL(신호)와 SLOT(함수)를 연결함

-위 함수 코드를 쓴 클래스 내부에서 '발생'하면 '발생하는 곳' 에 this 써도 됨.

- '발생하는 곳'은 포인터 형태여야 하지만, 대신 별명을 써도 됨 : '&발생하는곳' 형태로 입력

-시그널과 슬롯이 인자를 주고 받으면 다음과 같이 입력 : '함수이름(인자자료형)'

Ex)

```
connect(this,SIGNAL(reset_all()),m,SLOT(reset_all()));  
connect(m,SIGNAL(state(Point,int,int,int,int)),w,SLOT(state(Point,int,int,int,int)));
```

```
Q_SIGNALS:  
    void state(Point Goal, int degree, int R, int G, int B);  
  
public slots:  
    void state(Point Goal, int degree, int _R, int _G, int _B);  
  
Q_EMIT state(s.Goal, s.deg_theta, R, G, B);
```

```
Q_SIGNALS:  
    void reset_all();  
  
public slots:  
    void reset_all();  
  
Q_EMIT reset_all();
```

| 자주 쓰는 기능

QTimer

```
MainWindow::MainWindow(QWidget *parent) :  
    QMainWindow(parent),  
    ui(new Ui::MainWindow)  
{  
    ui->setupUi(this);  
    QTimer *timer=new QTimer;  
    timer->start();  
    timer->setInterval(10);  
    connect(timer,SIGNAL(timeout()),this,SLOT(update()));  
}
```

PaintEvent

```
private:  
    Ui::MainWindow *ui;  
    void paintEvent(QPaintEvent *e);
```

```
void MainWindow::paintEvent(QPaintEvent *e)  
{  
    QPainter painter(this);  
    QPen pen1(Qt::black);  
    QPoint p1, p2;  
    p1=j[0].sp;  
    p2=j[0].GetEndPoint(_degree);  
    pen1.setWidth(5);  
    painter.setPen(pen1);  
    painter.drawLine(p1,p2);  
}
```

| 빌드 및 실행

-빌드

cd 프로젝트 위치

mkdir build && cd build

cmake ..

make

-실행

./untitled : build 디렉토리 내에 실행파일 위치

과제 1

- 로봇 팔 만들기

원하는 각도에 맞게 관절을 수동 조절 가능하고,
자동으로 시계방향 또는 반시계방향으로 각 관절을 돌릴 수 있는 로봇 팔 UI 제작
필수 기능 및 조건)

각 관절 각도 수동 조정 기능

각관절 자동 반시계/시계방향 회전 기능

현재 로봇 팔의 상태(각 관절의 각도, 회전방향 등)를 txt 파일로 저장하는 기능

저장한 로봇 팔의 상태를 불러오는 기능

Class를 활용할 것

위 다섯 조건 외에 다른 건 모두 자유.

(기능의 구현 방식 등//원하면 다른 기능을 추가해도 상관x)

단, 사용자 편의성을 고려하여 만들 것

| 과제 2

- Queue 또는 stack 기능을 활용한 UI 제작

Queue : FIFO(First In First Out), 제일 처음 넣은 데이터가 처음으로 빠져나오는 구조

Stack : LIFO(Last In First Out), 제일 마지막에 넣은 데이터가 처음으로 빠져나오는 구조

벡터를 활용하여 구현할 것

과제

다음날 출근(7시) 전까지 예시 참고하여 GIT 제출

https://github.com/lee-sunkyoung/robit_intern_LSK

자신의 이름으로 패키지 생성후 개인톡으로 링크 제출

레포트 작성 시 **알아보기 쉽도록** 작성할 것.

- 목차 혹은 제목, 번호 붙이기 필수
- 코드 복사 붙여넣기금지
- 실행 결과 화면 첨부, 설명 작성 후pdf로 제출.

*** 제출 형식 틀리거나 기한 지키지 못한 과제는 사유불문 채점 안함.**