Qt Widget Quick Start

Blake Vermeer

April 3, 2017

Date Performed:	April 3, 2017
Company:	Keysight Technologies

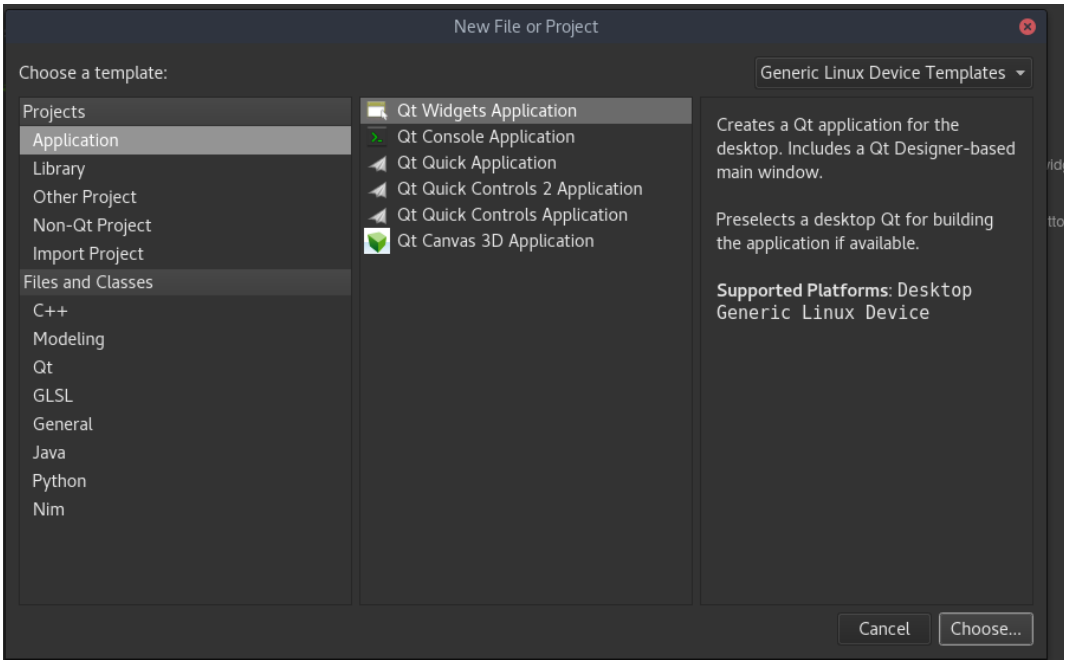
1 Overview

In this tutorial you will learn how to create a basic Qt Widget application from scratch and deploy it to the Keysight Hacking Platform development kit.

2 Creating a New Qt Widgets Project

1.) First we will use Qt Creator's new project wizard to create a new Qt Widgets project template. First open Qt Creator and then go to **File** → **New File or Project...**

2.) The new project wizard dialog box will pull up. Make sure the **Generic Linux Device Templates** option is selected in the drop-down menu in the upper right. Select **Applications** from the left hand menu under projects and then select the **Qt Widgets Application** project template and then click the **Choose...** button.



New File or Project

Choose a template: Generic Linux Device Templates

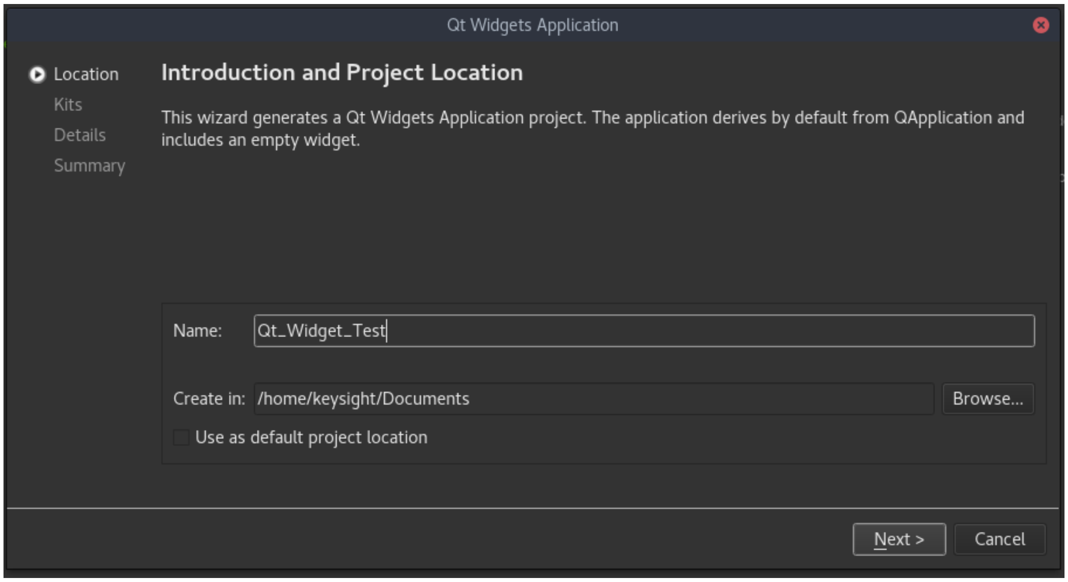
Creates a Qt application for the desktop. Includes a Qt Designer-based main window.

Preselects a desktop Qt for building the application if available.

Supported Platforms: Desktop Generic Linux Device

Figure 1: New Project Wizard Dialog Box

3.) In the next dialog box choose a name for the project and where to save it and then click the **Next** button.



Introduction and Project Location

This wizard generates a Qt Widgets Application project. The application derives by default from QApplication and includes an empty widget.

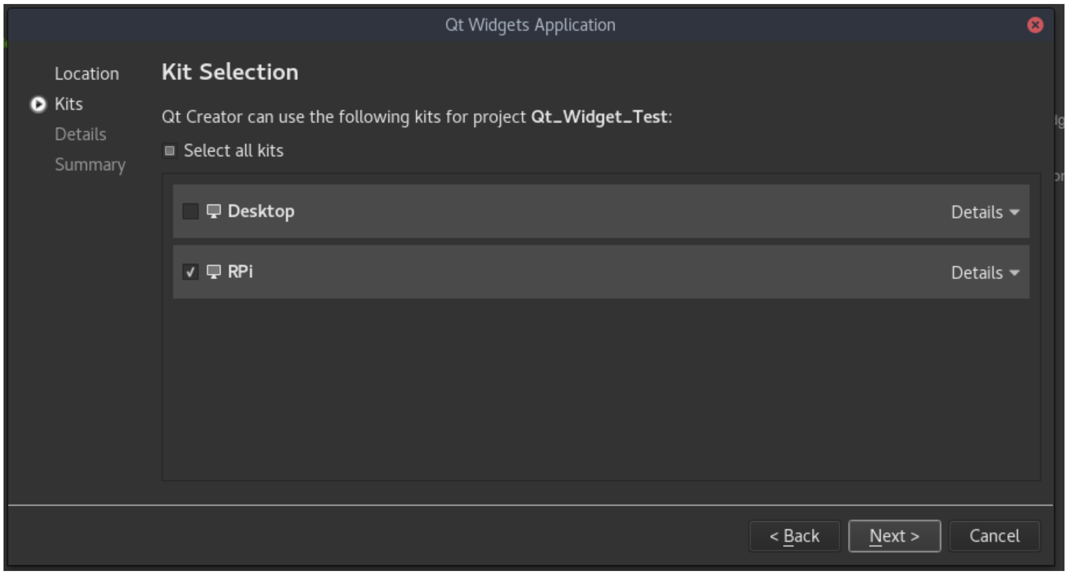
Name: Qt_Widget_Test

Create in: /home/keysight/Documents

☐ Use as default project location

Figure 2: Name the Project

4.) In the next dialog box make sure that only the **RPi** box is selected since we are not trying to build a desktop application and then click the **Next** button.



Kit Selection

Qt Creator can use the following kits for project **Qt_Widget_Test**:

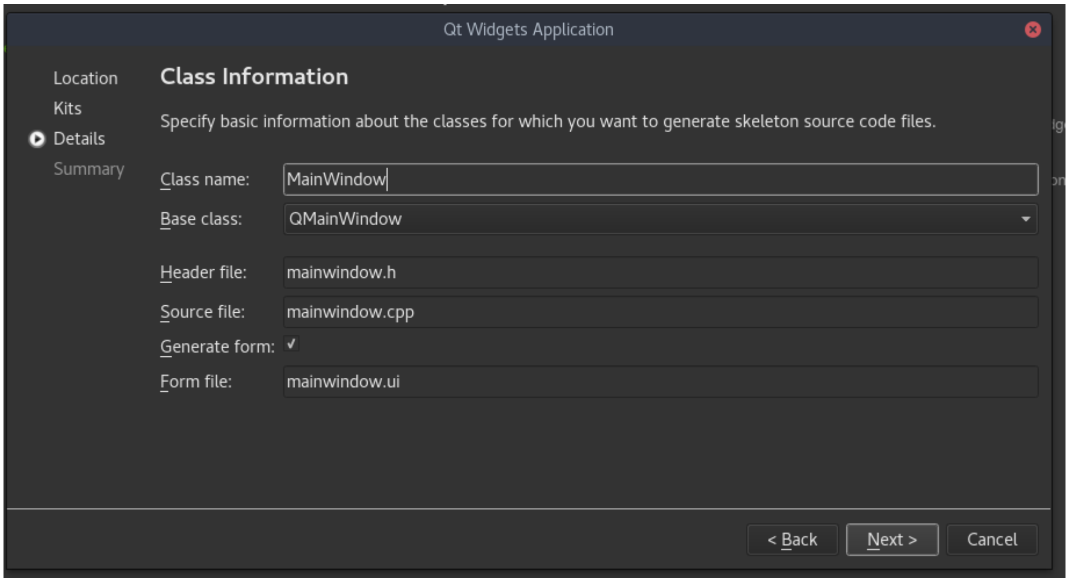
■ Select all kits

☐ Desktop

☑ RPi

Figure 3: Kit Selection

5.) In the **Class Information** dialog box leave everything at their default values and click the **Next** button.



Class Information

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

Class name: MainWindow

Base class: QMainWindow

Header file: mainwindow.h

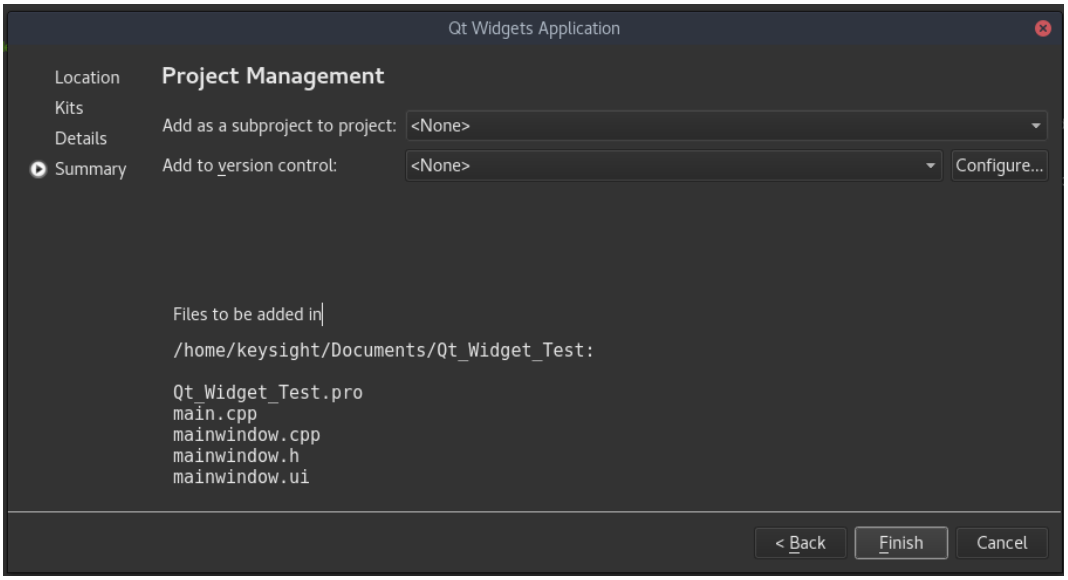
Source file: mainwindow.cpp

Generate form: ✓

Form file: mainwindow.ui

Figure 4: Class Information

6.) In the **Project Management** dialog box leave everything alone and click the **Finish** button.



Project Management

Add as a subproject to project: <None>

Add to version control: <None>

Files to be added in

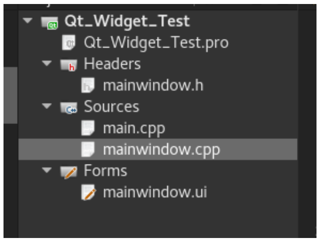
```
/home/keysight/Documents/Qt_Widget_Test:

Qt_Widget_Test.pro
main.cpp
mainwindow.cpp
mainwindow.h
mainwindow.ui
```

Figure 5: Project Management

3 Qt Widgets Project Overview

The Qt Widgets project wizard creates several different files. This section gives a brief overview of the files created and their purpose.



```
Qt_Widget_Test
    Qt_Widget_Test.pro
    Headers
        mainwindow.h
    Sources
        main.cpp
        mainwindow.cpp
    Forms
        mainwindow.ui
```

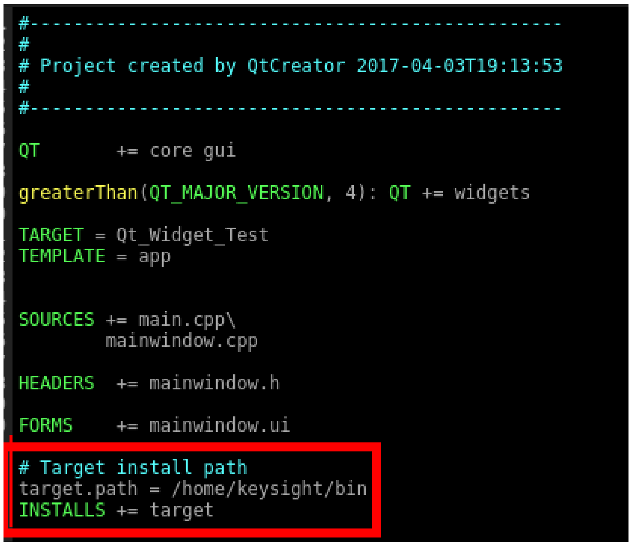
Figure 6: Project Files Created for a Project Name ”Qt_Widget_Test”

Here is a brief overview of the files created for a project called Qt_Widget_Test and their purpose:

- **Qt_Widget_Test.pro** - The main project file. The project file defines the source files used by the project, the name of the application, where to deploy the application on the target device, and various other settings.
- **mainwindow.h** - A header file for the *mainwindow* class.
- **main.cpp** - The main source file for the application. It creates a *QApplication* object need for Qt Widget programs and then creates a *mainwindow* object and displays it.
- **mainwindow.cpp** - The source file for the *mainwindow* class. This class inherits from *Q_OBJECT* and defines the actions done by the various *mainwindow* GUI events.
- **mainwindow.ui** - The UI for for the *mainwindow widget.* This file defines the GUI layout for the application.

4 Preparing the Project File

The default created project file is almost fully complete. The only thing that needs to be added is a target path definition that will tell Qt Creator where to install the application on the target device when deploying it. Add the three lines as shown in Figure 7.



```
#-------------------------------------------------
#
# Project created by QtCreator 2017-04-03T19:13:53
#
#-------------------------------------------------

QT       += core gui

greaterThan(QT_MAJOR_VERSION, 4): QT += widgets

TARGET = Qt_Widget_Test
TEMPLATE = app


SOURCES += main.cpp\
        mainwindow.cpp

HEADERS  += mainwindow.h

FORMS    += mainwindow.ui

# Target install path
target.path = /home/keysight/bin
INSTALLS += target
```

Figure 7: Define the target install path

5 Creating the GUI

This section will explain how to main a basic GUI. Double-click the *mainwindow.ui* file. It will automatically open in the design view.