Lecture 4. Introduction to Qt Libraries

Cross-Platform Application Development

October 13, 2017

Factors of Choice

Conveniences

- Substantial ease of the development process;
- Big number of built-in and trird-party components;
- Cross-platformness.

Inconveniences

- Installation and setup of the libraries;
- Distributing programs with library files;
- Lesser program running speed;
- Setting up projects for calling additional tools;
- Commercial license for additional functionality and tools.

Main Window Creation Example (Windows API)

Example

```
LRESULT CALLBACK WindowProcedure(HWND, UINT, WPARAM, LPARAM);

int APIENTRY WinMain(
  HINSTANCE hThisInstance, HINSTANCE hPrevInstance,
  LPSTR lpszArgument, int nCmdShow)
{
  WNDCLASSEX wnd_class;                          // window class
  wnd_class.lpszClassName = g_ctszClassName;     // class name
  wnd_class.lpfnWndProc = WindowProcedure;       // callback function
  // ...
  if (!RegisterClassEx(&wnd_class))              // class registration
    return -1;
  // ...
```

Main Window Creation Example (cont.)

Example (cont.)

```
HWND hWnd = CreateWindowEx(
  0,
  g_ctszClassName,      // window class name
  g_ctszAppTitle,
  WS_OVERLAPPEDWINDOW,
  CW_USEDEFAULT,
  CW_USEDEFAULT,
  CW_USEDEFAULT,
  CW_USEDEFAULT,
  HWND_DESKTOP,
  NULL,
  hThisInstance,
  NULL);
```

Main Window Creation Example (cont.)

Example (cont.)

```
  // Window displaying
  ShowWindow(hWnd, nCmdShow);
  // Message processing loop (until GetMessage() == 0)
  MSG messages;
  while (GetMessage(&messages, NULL, 0, 0))
  {
    // Sending the message to WindowProcedure()
    DispatchMessage(&messages);
  }
  // Returning a value passed to PostQuitMessage() to the system
  return messages.wParam;
}    // WinMain()
```

Main Window Creation Example (cont.)

Example (cont.)

```
LRESULT CALLBACK WindowProcedure(
  HWND hWnd, UINT uMesage, WPARAM wParam, LPARAM lParam)
{
  switch (uMesage)
  {
    case WM_MOUSEMOVE:
    {
      if (wParam & MK_LBUTTON && wParam & MK_SHIFT)
      {
        int nX = GET_X_LPARAM(lParam);
        int nY = GET_Y_LPARAM(lParam);
        // ...
      }
```

Main Window Creation Example (end)

Example (end)

```
      break;
    }    // case WM_MOUSEMOVE
    // case ...:
    case WM_DESTROY:
      PostQuitMessage(0);       // send WM_QUIT to the queue
      break;
    default:    // the rest messages are processed by the system
      return DefWindowProc(hWnd, uMesage, wParam, lParam);
  }    // switch (uMesage)
  //
  return 0;
}    // WindowProcedure()
```

Project Structure

```
⟨working_directory⟩
└─ 01-hello  ..... project source files directory
   ├─ example-01.cpp
   └─ CMakeLists.txt
```

Figure 1: structure for the simple project directory

Example

Example (CMakeLists.txt)

```
cmake_minimum_required(VERSION 2.8.11)

project(example-01)

find_package(Qt5Widgets)

add_executable(example-01 example-01.cpp)

target_link_libraries(example-01 Qt5::Widgets)
```

Example of the Main Window Use

Example (example-01.cpp)

```
#include <QApplication>
#include <QLabel>

int main(int nArgC, char *apszArgV[])
{
  QApplication app(nArgC, apszArgV);
  QLabel *pLabel = new QLabel("Hello Qt!");
  pLabel->show();
  //
  return app.exec();
}
```

Project Building

⟨build_directory⟩
- build_01-hello
  - build.cmd
  - libgcc_s_dw2-1.dll, libstdc++-6.dll, libwinpthread-1.dll, Qt5Core.dll, Qt5Gui.dll, Qt5Widgets.dll ..... *copied from* ⟨*Qt_path*⟩\bin
  - example-01.cbp ..... *generated on running* build.cmd
  - ...
  - example-01.exe ..... *generated on the project build*

Figure 2: structure for the project build directory

Example (cont.)

Example (build.cmd)

```
set PATH=C:\Qt\Qt5.7.0\Tools\mingw530_32\bin;%PATH%

cmake^
  -G "CodeBlocks - MinGW Makefiles"^
  -D CMAKE_PREFIX_PATH="C:\Qt\Qt5.7.0\5.7\mingw53_32"^
   ⟨path_to_projects⟩\01-hello
```

Example (cont.)

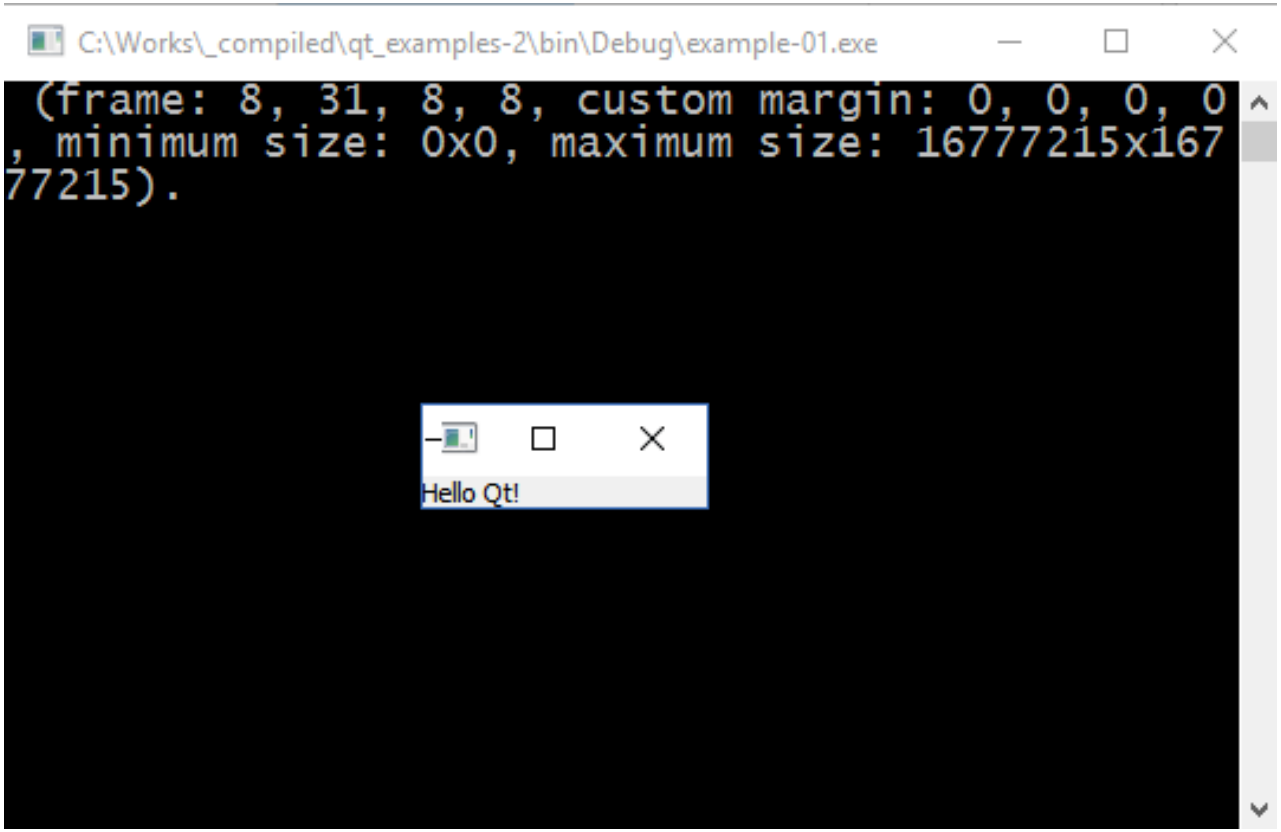

Figure 3: Qt application window

Example (end)

Example (CMakeLists.txt)

```
cmake_minimum_required(VERSION 2.8.11)

# ...

add_executable(example-01 WIN32 example-01.cpp)

target_link_libraries(example-01 Qt5::Widgets)
```

Qt Classes Hierarchy

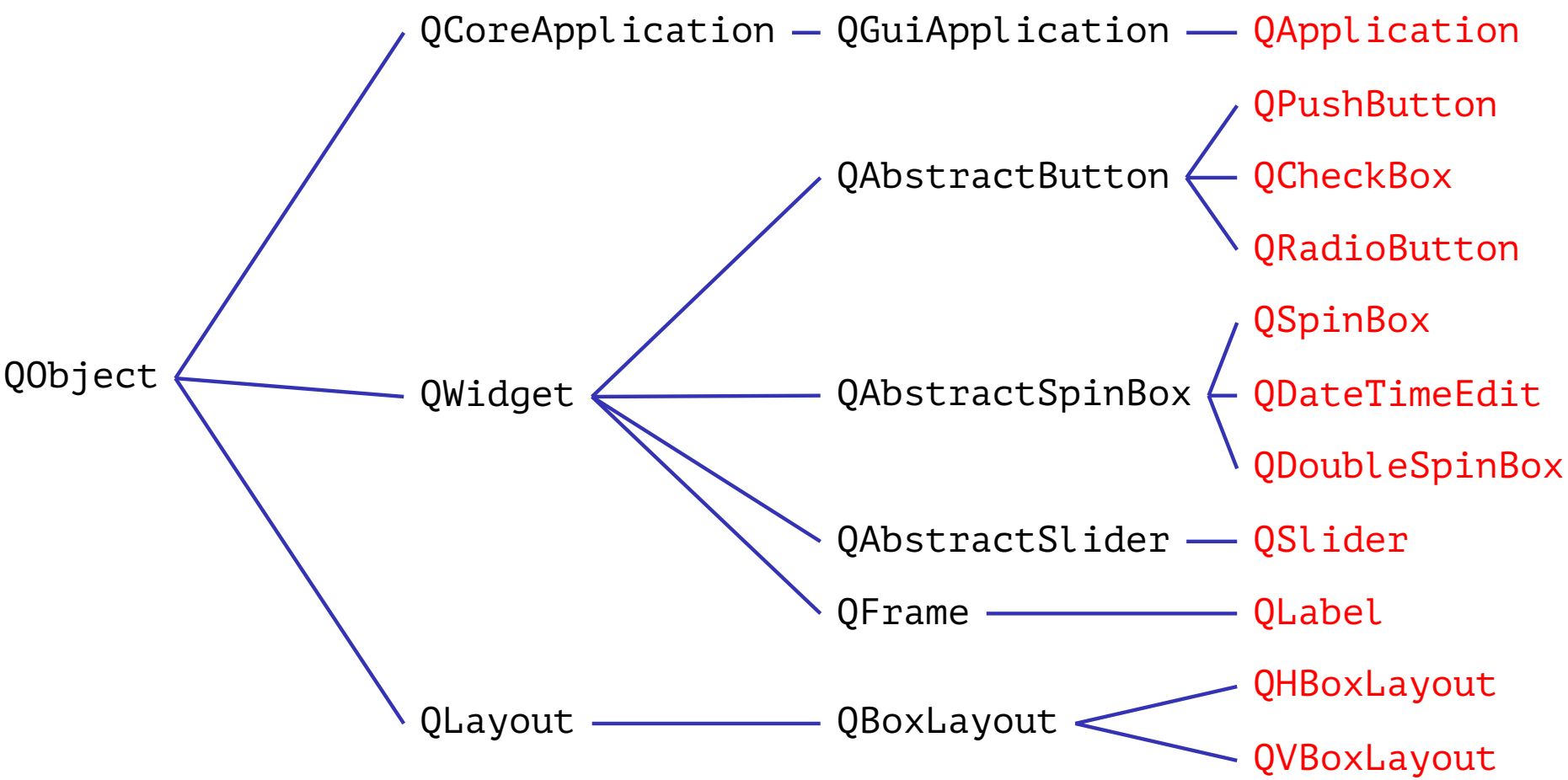

Figure 4: part of Qt class hierarchy

Object Joining

Callback Functions (Old Approach)

- Type unsafety;
- "1:1" connection between the processing function and callback function.

Definitions

Signal: *emitted* on some given event;

Slot: a function called in response to the given signal.

Example

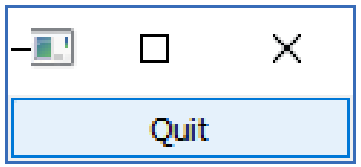

Figure 5: an application window with a button as a main window

Example (end)

Example (example-02.cpp)

```
#include <QApplication>
#include <QPushButton>

int main(int nArgC, char *apszArgV[])
{
  QApplication app(nArgC, apszArgV);
  QPushButton *pButton = new QPushButton("&Quit");
  QObject::connect(pButton, SIGNAL(clicked()), &app, SLOT(quit()));
  pButton->show();
  //
  return app.exec();
}
```

QAbstractButton Signals and QCoreApplication Slots

QAbstractButton Signals

- **void** clicked(**bool** checked = **false**)
- **void** pressed()
- **void** released()
- **void** toggled(**bool** checked)

QCoreApplication Slots

- **void** quit()

QWidget Slots

QWidget Slots

- **bool** close()
- **void** hide()
- **void** lower()
- **void** raise()
- **void** repaint()
- **void** setEnabled(**bool**)
- **void** setFocus()

QWidget Slots (end)

- **void** setWindowTitle(**const** QString &)
- **void** show()
- **void** showFullScreen()
- **void** showMaximized()
- **void** showMinimized()
- **void** showNormal()

Example of UI Elements Interaction

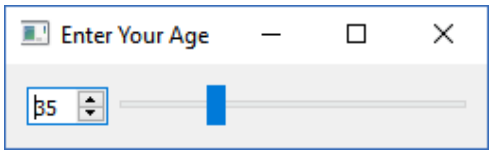

Figure 6: an application window with two UI elements

QSpinBox Signals and QAbstractSlider Slots

QSpinBox Signals

- **void** valueChanged(**int** i)
- **void** valueChanged(**const** QString &text)

QAbstractSlider Slots

- **void** setOrientation(Qt::Orientation)
- **void** setValue(**int**)

QAbstractSlider Signals and QSpinBox Slots

QAbstractSlider Signals

- **void** actionTriggered(**int** action)
- **void** rangeChanged(**int** min, **int** max)
- **void** sliderMoved(**int** value)
- **void** sliderPressed()
- **void** sliderReleased()
- **void** valueChanged(**int** value)

QSpinBox Slots

- **void** setValue(**int** val)

Example of the Layout Use

Example (example-03.cpp)

```
#include <QApplication>
#include <QHBoxLayout>
#include <QSlider>
#include <QSpinBox>

int main(int nArgC, char *apszArgV[])
{
  QApplication app(nArgC, apszArgV);
  QWidget *pWindow = new QWidget;
  pWindow->setWindowTitle("Enter Your Age");
  QSpinBox *pSpinBox = new QSpinBox;
  QSlider *pSlider = new QSlider(Qt::Horizontal);
```

Example of the Layout Use (cont.)

Example (example-03.cpp, cont.)

```
pSpinBox->setRange(0, 130);
pSlider->setRange(0, 130);
QObject::connect(
  pSpinBox, SIGNAL(valueChanged(int)), pSlider, SLOT(setValue(int)));
QObject::connect(
  pSlider, SIGNAL(valueChanged(int)), pSpinBox, SLOT(setValue(int)));
pSpinBox->setValue(35);
//
QHBoxLayout *pLayout = new QHBoxLayout;
pLayout->addWidget(pSpinBox);
pLayout->addWidget(pSlider);
pWindow->setLayout(pLayout);
```

Example of the Layout Use (end)

Example (example-03.cpp, end)

```
    pWindow->show();
    //
    return app.exec();
    //
}     // main()
```

Signals and Slots Use

Syntax

```
QObject::connect(
    ⟨pointer_to_sender⟩, SIGNAL(⟨signal_name⟩(⟨arguments⟩)),
    ⟨pointer_to_receiver⟩, SLOT(⟨slot_name⟩(⟨arguments⟩)));
```

Rules

- Connected signals and slots must have the same types of parameters ("extra" parameters of signals are ignored).
- One signal can be connected to several slots.
- Several signals can be connected to one slot.
- Signals can be connected to signals.
- The connections can be removed (`QObject::disconnect()`).

Introspection

Definitions

Metaprogram: a program which generates or modifies other programs (for example, itself).

(Type) Introspection: the ability in a programming language to detect the object type and structure at the program run time.

Generating Metainformation

Example (myobject.h)

```
#include <QObject>

class MyObject : public QObject
{
  Q_OBJECT
  //
```

Example (myobject.h, end)

```
  // ...
private slots:
  //
  void slot1();
};    // class MyObject
```

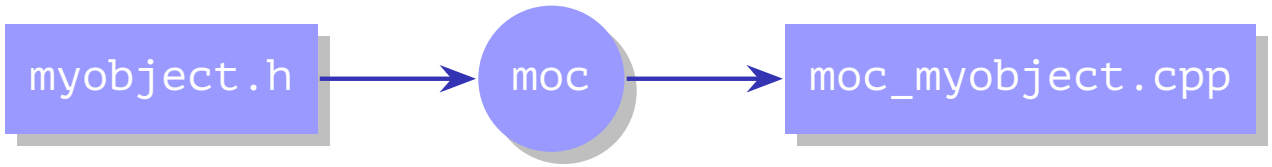

Figure 7: generating metainformation with moc tool

Counter Example

Example (counter.h)

```
#include <QObject>

class Counter : public QObject
{
  Q_OBJECT
  //
public:
  Counter()
    : m_nValue(0) { }
  int value() const
    { return m_nValue; }
```

Example (counter.h, end)

```
public slots:
  void setValue(int nValue);
signals:
  void valueChanged(int nValue);
private:
  int m_nValue;
};    // class Counter
```

Counter Example (cont.)

Example (counter.cpp)

```
#include "counter.h"

void Counter::setValue(int nValue)
{
  if (nValue != m_nValue)
  {
    m_nValue = nValue;
    emit valueChanged(nValue);
  }
}
```

Counter Example (end)

Example (main.cpp)

```
#include "counter.h"

int main()
{
  Counter a, b;
  QObject::connect(
    &a, &Counter::valueChanged,
    &b, &Counter::setValue);
  //
  a.setValue(12);     // a.value() == 12, b.value() == 12
  b.setValue(48);     // a.value() == 12, b.value() == 48
}
```

Example

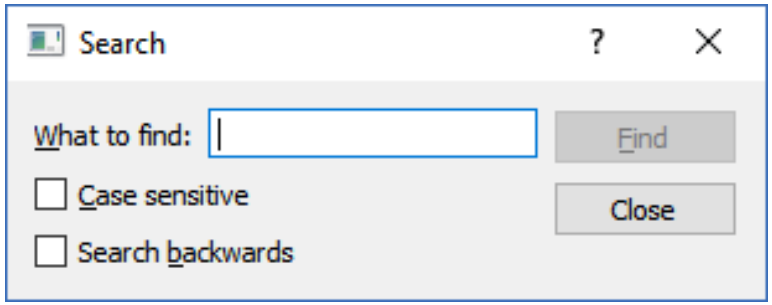

Figure 8: an application window based on `QDialog` class

Example (cont.)

```
⟨working_directory⟩
└── 04-dialog
    ├── find-dlg.cpp  ..... FindDialog class methods
    ├── find-dlg.h
    └── ...
```

Figure 9: structure for the project using forms

```
⟨build_directory⟩
└── build_04-dialog
    ├── ...
    └── moc_find-dlg.cpp  ..... generated with moc from find-dlg.h
```

Figure 10: structure for the project build directory

Example (cont.)

Example (find-dlg.h)

```
#ifndef FIND_DLG_H__
#define FIND_DLG_H__

#include <QDialog>

class QCheckBox;
class QLabel;
class QLineEdit;
class QPushButton;
```

Example (find-dlg.h, cont.)

```
class FindDialog : public QDialog
{
  Q_OBJECT
  //
public:
  //
  FindDialog(QWidget *pParent = 0);
  //
```

Example (cont.)

Example (find-dlg.h, cont.)

```
signals:
  //
  void findNext(
    const QString &rcStr, Qt::CaseSensitivity nCaseSensitivity);
  void findPrev(
    const QString &rcStr, Qt::CaseSensitivity nCaseSensitivity);
  //
private slots:
  //
  void findClicked();
  void enableButtonFind(const QString &rcText);
  //
```

Example (cont.)

Example (find-dlg.h, end)

```
private:
  //
  QLabel *m_pLabel;
  QLineEdit *m_pLineEdit;
  QCheckBox *m_pCheckBoxCase;
  QCheckBox *m_pCheckBoxBack;
  QPushButton *m_pButtonFind;
  QPushButton *m_pButtonClose;
};    // class FindDialog

#endif    // FIND_DLG_H__
```

Example (cont.)

Example (find-dlg.cpp)

```
#include "find-dlg.h"

#include <QtWidgets>

FindDialog::FindDialog(QWidget *pParent)
  : QDialog(pParent)
{
  m_pLabel = new QLabel(
    QString::fromLocal8Bit("&What to find:"));    // fromUtf8(...)
  m_pLineEdit = new QLineEdit;
  m_pLabel->setBuddy(m_pLineEdit);
  //
```

Example (cont.)

Example (find-dlg.cpp, cont.)

```
m_pCheckBoxCase = new QCheckBox(
  QString::fromLocal8Bit("&Case sensitive"));
m_pCheckBoxBack = new QCheckBox(
  QString::fromLocal8Bit("Search &backwards"));
//
m_pButtonFind = new QPushButton(
  QString::fromLocal8Bit("&Find"));
m_pButtonFind->setDefault(true);
m_pButtonFind->setEnabled(false);
//
m_pButtonClose = new QPushButton(
  QString::fromLocal8Bit("Close"));
//
```

Example (cont.)

Example (find-dlg.cpp, cont.)

```
connect(
  m_pLineEdit, SIGNAL(textChanged(const QString &)),
  this, SLOT(enableButtonFind(const QString &)));
connect(
  m_pButtonFind, SIGNAL(clicked()),
  this, SLOT(findClicked()));
connect(
  m_pButtonClose, SIGNAL(clicked()),
  this, SLOT(close()));
//
QHBoxLayout *pLayoutTopLeft = new QHBoxLayout;
pLayoutTopLeft->addWidget(m_pLabel);
pLayoutTopLeft->addWidget(m_pLineEdit);
```

Example (cont.)

Example (find-dlg.cpp, cont.)

```
//
QVBoxLayout *pLayoutLeft = new QVBoxLayout;
pLayoutLeft->addLayout(pLayoutTopLeft);
pLayoutLeft->addWidget(m_pCheckBoxCase);
pLayoutLeft->addWidget(m_pCheckBoxBack);
//
QVBoxLayout *pLayoutRight = new QVBoxLayout;
pLayoutRight->addWidget(m_pButtonFind);
pLayoutRight->addWidget(m_pButtonClose);
pLayoutRight->addStretch();
//
```

Example (cont.)

Example (find-dlg.cpp, cont.)

```
  QHBoxLayout *pLayoutMain = new QHBoxLayout;
  pLayoutMain->addLayout(pLayoutLeft);
  pLayoutMain->addLayout(pLayoutRight);
  setLayout(pLayoutMain);
  //
  setWindowTitle(
    QString::fromLocal8Bit("Search"));
  setFixedHeight(sizeHint().height());
}    // FindDialog::FindDialog()
```

Example (cont.)

Example (find-dlg.cpp, cont.)

```
void FindDialog::findClicked()
{
  const QString cText = m_pLineEdit->text();
  const Qt::CaseSensitivity cnCaseSensitivity =
    m_pCheckBoxCase->isChecked() ?
    Qt::CaseSensitive :
    Qt::CaseInsensitive;
  //
  if (m_pCheckBoxBack->isChecked())
    emit findPrev(cText, cnCaseSensitivity);
  else
    emit findNext(cText, cnCaseSensitivity);
}
```

Example (cont.)

Example (find-dlg.cpp, end)

```
void FindDialog::enableButtonFind(const QString &rcText)
{
  m_pButtonFind->setEnabled(!rcText.isEmpty());
}

// End of File
```

Example (cont.)

Example (example-04.cpp)

```
#include "find-dlg.h"

#include <QApplication>

int main(int nArgC, char *apszArgV[])
{
  QApplication app(nArgC, apszArgV);
  FindDialog *pDialog = new FindDialog;
  pDialog->show();
  //
  return app.exec();
}
```

Example (end)

Example (CMakeLists.txt)

```
# ...

qt5_wrap_cpp(MOC_WRAPPERS find-dlg.h)

# ...

add_executable(
  example-04 WIN32
  example-04.cpp find-dlg.cpp find-dlg.h
  ${MOC_WRAPPERS})

target_link_libraries(example-04 Qt5::Widgets)
```

Visual Form Editor

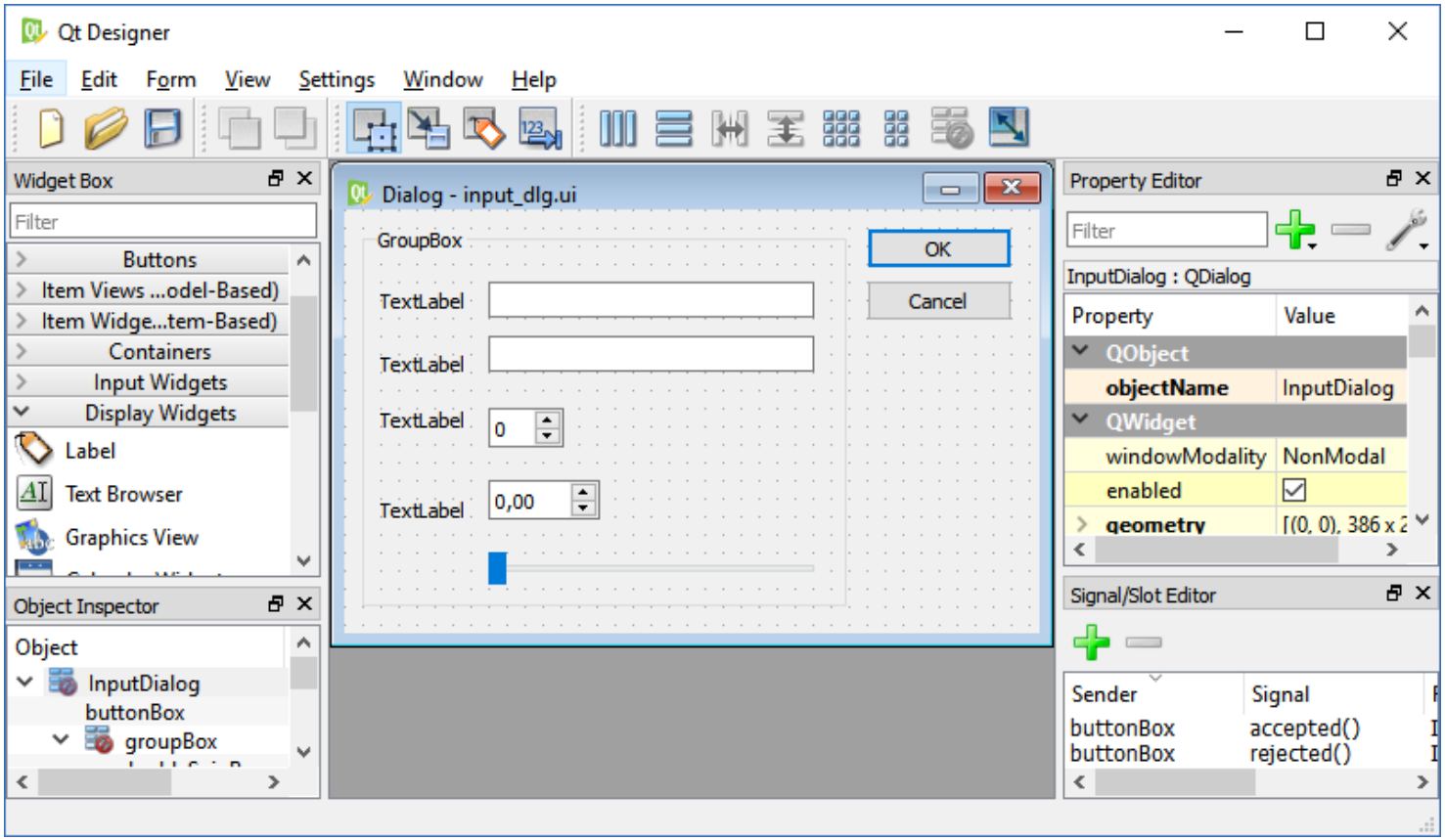

Figure 11: a window of Qt Designer form editor

Generated Code from the Form Description

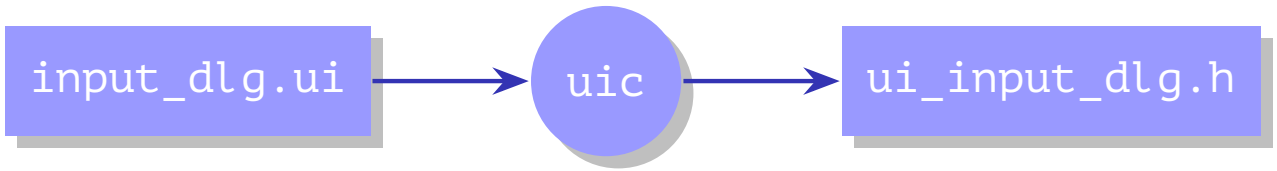

Figure 12: generating UI description into code on C++ with `uic` tool

Generating Code from a Dialog Description

Example (input_dlg.ui)

```
<?xml version="1.0"
  encoding="UTF-8"?>
<ui version="4.0">
 <class>InputDialog</class>
 <widget class="QDialog"
   name="InputDialog">
  <property name="geometry">
   <rect>
    <x>0</x>
    <y>0</y>
    <width>386</width>
    <height>235</height>
    <!-- ... -->
```

Example (ui_input_dlg.h)

```
class Ui_InputDialog
{
public:
  QDialogButtonBox *buttonBox;
  QGroupBox *groupBox;
  QLabel *label_2;
  QLineEdit *lineEdit_2;
  // ...
  void setupUi(QDialog *InputDialog)
  {
    // ...
    InputDialog->resize(386, 235);
    // ...
```

Example

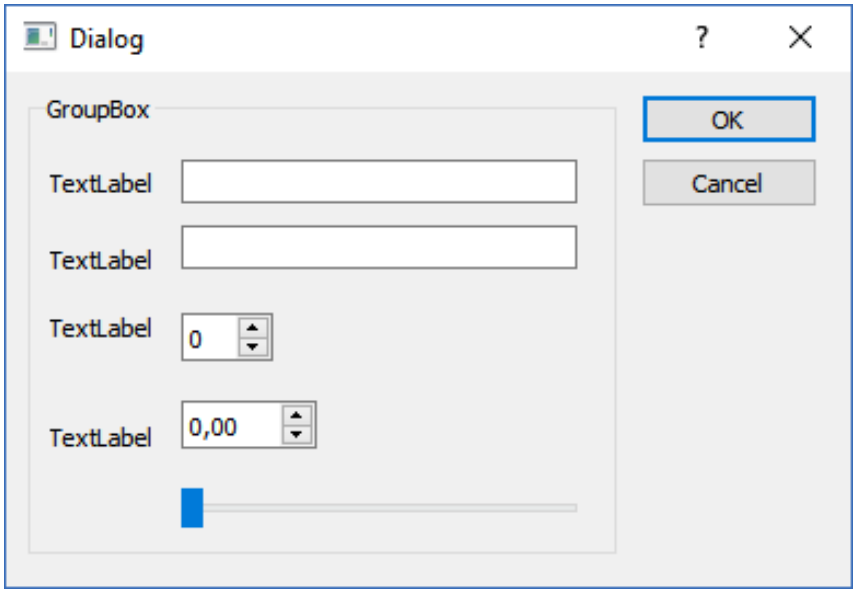

Figure 13: main window for the dialog-based application

Example (cont.)

⟨working_directory⟩
- 05-ui
  - example-05.cpp
  - CMakeLists.txt
  - input_dlg.ui ..... *created/edited with Qt Designer*

Figure 14: structure for the directory of the project that uses forms

⟨build_directory⟩
- build_05-ui
  - ...
  - ui_input_dlg.h ..... *generated with* `uic` *from* `input_dlg.ui`

Figure 15: structure for the project build directory

Example (cont.)

Example (CMakeLists.txt)

```
# ...

qt5_wrap_ui(UIC_WRAPPERS input_dlg.ui)

add_executable(
  example-05 WIN32
  example-05.cpp ${UIC_WRAPPERS})

target_link_libraries(example-05 Qt5::Widgets)
```

Example (end)

Example (example-05.cpp)

```
#include "ui_input_dlg.h"
#include <QApplication>
#include <QDialog>

int main(int nArgC, char *apszArgV[])
{
  QApplication app(nArgC, apszArgV);
  Ui::InputDialog ui;
  QDialog *pDialog = new QDialog;
  ui.setupUi(pDialog);
  pDialog->show();
  return app.exec();
}
```

Connecting Signals to Slots

Ways of Connecting

- Programmatically.
- In the form designer.
- Automatically (`setupUi()`).

Example

```
QObject::connect(
  pButton, SIGNAL(clicked()),
  &app, SLOT(quit()));
```

Connecting Signals to Slots

Ways of Connecting

- Programmatically.
- In the form designer.
- Automatically (`setupUi()`).

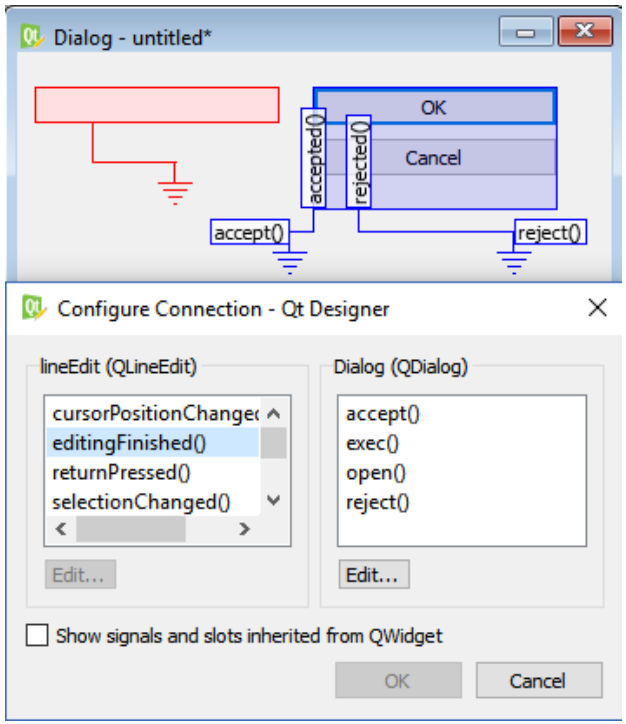

Figure 16: connection editor

Connecting Signals to Slots

Ways of Connecting

- Programmatically.
- In the form designer.
- Automatically (`setupUi()`).

Slot Naming

on_⟨*object_name*⟩_⟨*signal_name*⟩(⟨*parameters*⟩);

Example

⟨working_directory⟩
- 06-subclass
  - example-06.cpp
  - input-dlg.cpp
  - input-dlg.h ……QDialog *descendant*
  - CMakeLists.txt
  - input_dlg.ui …*as before (Qt Designer)*

Figure 16: structure for the directory of the project with interactive forms

⟨build_directory⟩
- build_06-subclass
  - …
  - example-05.cbp
  - moc_input-dlg.cpp *generated with* moc *from* input-dlg.h
  - ui_input_dlg.h *as before*

Figure 17: structure for the project build directory

Example (cont.)

Example (CMakeLists.txt)

```
cmake_minimum_required(VERSION 2.8.11)

project(example-06)

find_package(Qt5Widgets)

set(CMAKE_INCLUDE_CURRENT_DIR ON)

qt5_wrap_cpp(MOC_WRAPPERS input-dlg.h)
qt5_wrap_ui(UIC_WRAPPERS input_dlg.ui)
```

Example (cont.)

Example (CMakeLists.txt, end)

```
add_executable(
  example-06 WIN32
  example-06.cpp input-dlg.cpp input-dlg.h
  ${MOC_WRAPPERS} ${UIC_WRAPPERS})

target_link_libraries(example-06 Qt5::Widgets)

# End of File
```

Example (cont.)

Example (input-dlg.h)

```
#ifndef INPUT_DLG_H__
#define INPUT_DLG_H__

#include "ui_input_dlg.h"

#include <QDialog>

class InputDialog : public QDialog, public Ui::InputDialog
{
  Q_OBJECT
  //
```

Example (cont.)

Example (input-dlg.h, end)

```
public:
  //
  InputDialog(QWidget *pParent = 0);
  //
private slots:
  //
  void on_lineEdit_textChanged();
};    // class InputDialog

#endif    // INPUT_DLG_H__
```

Example (cont.)

Example (input-dlg.cpp)

```
#include "input-dlg.h"

#include <QtWidgets>

InputDialog::InputDialog(QWidget *pParent)
  : QDialog(pParent)
{
  setupUi(this);
  //
  connect(
    lineEdit_2, SIGNAL(textChanged(const QString &)),
    this, SLOT(on_lineEdit_textChanged()));
  //
```

Example (cont.)

Example (input-dlg.cpp, end)

```
  on_lineEdit_textChanged();
  //
}    // InputDialog::InputDialog()

void InputDialog::on_lineEdit_textChanged()
{
  QPushButton *pButton = buttonBox->button(
    QDialogButtonBox::Ok);
  pButton->setEnabled(
    !lineEdit->text().isEmpty() &&
    !lineEdit_2->text().isEmpty());
}
```

Example (cont.)

Example (example-06.cpp)

```
#include "input-dlg.h"

#include <QApplication>
#include <QMessageBox>

int main(int nArgC, char *apszArgV[])
{
  QApplication app(nArgC, apszArgV);
  InputDialog *pDialog = new InputDialog;
  // pDialog->show();
  const int cnResult = pDialog->exec();
```

Example (cont.)

Example (example-06.cpp, cont.)

```
if (cnResult == QDialog::Accepted)
{
  QString message =
    QString::fromLocal8Bit("Result %1 %2: %3 %4 %5").
      arg(pDialog->lineEdit->text()).
      arg(pDialog->lineEdit_2->text()).
      arg(pDialog->spinBox->value()).
      arg(pDialog->doubleSpinBox->value()).
      arg(pDialog->horizontalSlider->value());
  //
```

Example (end)

Example (example-06.cpp, end)

```
    QMessageBox::information(
      0, QString::fromLocal8Bit("Result"), message);
    //
  }    // if (cnResult == QDialog::Accepted)
  //
  // return app.exec();
  //
}    // main()
```