Lecture 6. Implementing User Interface Elements in Qt

Cross-Platform Application Development

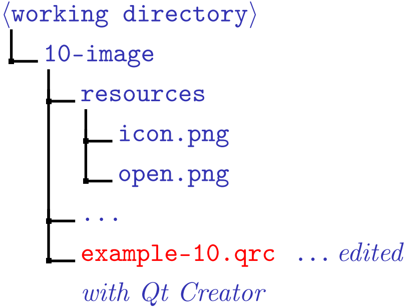
November 3, 2017

Contents Overview

Used Facilities

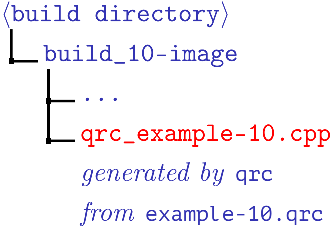
- Resources;
- The main application window with a menu and panels;
- Dynamic menus;
- Saving application settings to a system-independent storage;
- Window scrolling;
- Localization;
- Application installing.

Project Structure



⟨working directory⟩
└ 10-image
  ├ resources
  │ ├ icon.png
  │ └ open.png
  ├ ...
  └ example-10.qrc ... *edited with Qt Creator*

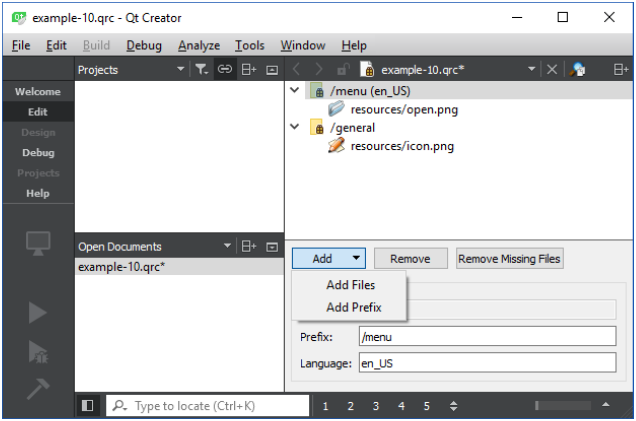
Figure 1: a structure for the project directory using resources



⟨build directory⟩
└ build_10-image
  ├ ...
  └ qrc_example-10.cpp *generated by* qrc *from* example-10.qrc

Figure 2: a structure for the build directory

Resource Editor


example-10.qrc - Qt Creator

/menu (en_US)
resources/open.png
/general
resources/icon.png

Prefix: /menu

Language: en_US

Figure 3: resource editing with Qt Creator

Example

Example (example-10.qrc)

```
<RCC>
    <qresource prefix="/menu" lang="en_US">
        <file>resources/open.png</file>
    </qresource>
    <qresource prefix="/general">
        <file>resources/icon.png</file>
    </qresource>
</RCC>
```

Attaching Resources in the Project

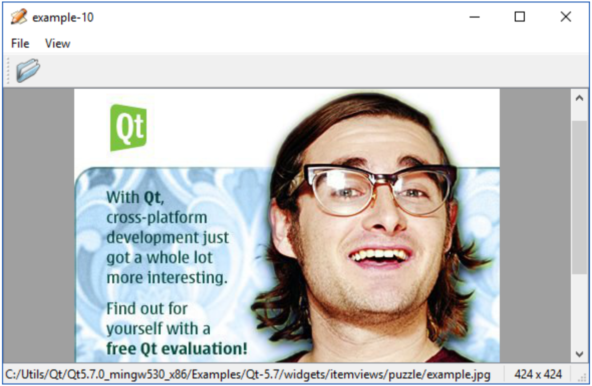
Example (CMakeLists.txt)

```
# ...

qt5_add_resources(
  QRC_WRAPPERS
  example-10.qrc
  OPTIONS -no-compress)

# ...
```

Example (CMakeLists.txt, end)

```
add_executable(
  example-10 WIN32
  ${CPP_FILES}
  ${MOC_WRAPPERS}
  ${QRC_WRAPPERS})

# ...
```

Example


example-10
File
View
Qt
With Qt, cross-platform development just got a whole lot more interesting.
Find out for yourself with a free Qt evaluation!

C:/Utils/Qt/Qt5.7.0_mingw530_x86/Examples/Qt-5.7/widgets/itemviews/puzzle/example.jpg

424 x 424

Figure 4: a main window with control elements

Example

Example (main-window.h)

```
#ifndef MAIN_WINDOW_H__
#define MAIN_WINDOW_H__

#include <QMainWindow>
#include <QSize>
#include <QStringList>

class QLabel;
class QScrollArea;
class QAction;
```

Example (cont.)

Example (main-window.h, cont.)

```
class MainWindow : public QMainWindow
{
  Q_OBJECT
  //
public:
  //
  MainWindow();
  //
protected:
  //
  virtual void closeEvent(QCloseEvent *pEvent);
  //
```

Example (cont.)

Example (main-window.h, cont.)

```
private slots:
  //
  void open();
  void openRecentFile();
  //
private:
  //
  QLabel *m_pLabelImage;
  QLabel *m_pLabelFileName;
  QLabel *m_pLabelImageSize;
  QScrollArea *m_pAreaImage;
  //
```

Example (cont.)

Example (main-window.h, cont.)

```
enum
{
  enMaxRecentFiles = 5
};
//
QAction *m_apActionsRecent[enMaxRecentFiles];
QAction *m_pActionSeparator;
QAction *m_pActionViewFile;
//
```

Example (cont.)

Example (main-window.h, end)

```
  QString m_fileName;
  QSize m_sizeImage;
  QStringList m_listRecentFiles;
  //
  void loadFile(const QString &rcFileName);
  void updateRecentFileActions();
  void updateStatusBar();
};    // class MainWindow

#endif    // MAIN_WINDOW_H__
```

Example (cont.)

Example (main-window.cpp)

```
#include "main-window.h"

#include <QtWidgets>

namespace
{
  QString strippedName(const QString &rcFullFileName)
  {
    return QFileInfo(rcFullFileName).fileName();
  }
}
```

Example (cont.)

Example (main-window.cpp, cont.)

```cpp
MainWindow::MainWindow()
{
  m_pLabelImage = new QLabel;
  m_pLabelImage->setAlignment(Qt::AlignCenter);
  //
  m_pAreaImage = new QScrollArea;
  m_pAreaImage->setBackgroundRole(QPalette::Dark);
  m_pAreaImage->setWidget(m_pLabelImage);
  m_pAreaImage->setWidgetResizable(true);
  setCentralWidget(m_pAreaImage);
  //
```

Example (cont.)

Example (main-window.cpp, cont.)

```
QMenu *pMenuFile = menuBar()->addMenu(tr("&File"));
QAction *pActionFile = pMenuFile->addAction(
  QIcon(":/menu/resources/open.png"),
  tr("&Open"),
  this,
  SLOT(open()),
  QKeySequence::Open);
m_pActionSeparator = pMenuFile->addSeparator();
m_pActionSeparator->setVisible(false);
```

Example (cont.)

Example (main-window.cpp, cont.)

```
for (int i = 0; i < enMaxRecentFiles; ++ i)
{
  m_apActionsRecent[i] = new QAction(this);
  m_apActionsRecent[i]->setVisible(false);
  connect(
    m_apActionsRecent[i], SIGNAL(triggered()),
    this, SLOT(openRecentFile()));
  pMenuFile->addAction(m_apActionsRecent[i]);
}
```

Example (cont.)

Example (main-window.cpp, cont.)

```
pMenuFile->addSeparator();
QAction *pActionExit = pMenuFile->addAction(
  tr("E&xit"),
  this,
  SLOT(close()),
  QKeySequence::Quit);
//
QToolBar *pToolBarFile = addToolBar(tr("&File"));
pToolBarFile->addAction(pActionFile);
//
```

Example (cont.)

Example (main-window.cpp, cont.)

```
QMenu *pMenuView = menuBar()->addMenu(tr("&View"));
m_pActionViewFile = pMenuView->addAction(
  tr("File &tool bar"),
  pToolBarFile,
  SLOT(setVisible(bool)));
m_pActionViewFile->setCheckable(true);
//
m_pAreaImage->addAction(pActionFile);
m_pAreaImage->addAction(pActionExit);
m_pAreaImage->setContextMenuPolicy(
  Qt::ActionsContextMenu);
//
```

Example (cont.)

Example (main-window.cpp, cont.)

```
m_pLabelFileName = new QLabel;
m_pLabelImageSize = new QLabel("9999 x 9999");
m_pLabelImageSize->setAlignment(Qt::AlignHCenter);
m_pLabelImageSize->setMinimumSize(
  m_pLabelImageSize->sizeHint());
//
statusBar()->addWidget(m_pLabelFileName, 1);
statusBar()->addWidget(m_pLabelImageSize);
updateStatusBar();
//
setWindowIcon(QIcon(":/general/resources/icon.png"));
//
```

Example (cont.)

Example (main-window.cpp, cont.)

```
  QSettings settings("stu003", "example-10");
  restoreGeometry(
    settings.value("geometry").toByteArray());
  m_listRecentFiles =
    settings.value("recentFiles").toStringList();
  updateRecentFileActions();
  const bool cbShowFileTools =
    settings.value("showFileTools", true).toBool();
  m_pActionViewFile->setChecked(cbShowFileTools);
  pToolBarFile->setVisible(cbShowFileTools);
}    // MainWindow::MainWindow()
```

Example (cont.)

Example (main-window.cpp, cont.)

```
void MainWindow::closeEvent(QCloseEvent *pEvent)
{
  pEvent->accept();    // ignore()
  //
  QSettings settings("stu003", "example-10");
  settings.setValue("geometry", saveGeometry());
  settings.setValue("recentFiles", m_listRecentFiles);
  settings.setValue(
    "showFileTools", m_pActionViewFile->isChecked());
}
```

Example (cont.)

Example (main-window.cpp, cont.)

```
void MainWindow::open()
{
  QString fileName = QFileDialog::getOpenFileName(this);
  if (!fileName.isEmpty())
    loadFile(fileName);
}

void MainWindow::openRecentFile()
{
  QAction *pAction = qobject_cast <QAction *> (sender());
  if (pAction)
    loadFile(pAction->data().toString());
}
```

Example (cont.)

Example (main-window.cpp, cont.)

```
void MainWindow::loadFile(const QString &rcFileName)
{
  QPixmap pixmap;
  QApplication::setOverrideCursor(Qt::WaitCursor);
  const bool cbSuccess = pixmap.load(rcFileName);
  QApplication::restoreOverrideCursor();
  //
```

Example (cont.)

Example (main-window.cpp, cont.)

```
  if (cbSuccess)
  {
    m_pLabelImage->setPixmap(pixmap);
    m_sizeImage = pixmap.rect().size();
    m_fileName = rcFileName;
    m_listRecentFiles.removeAll(m_fileName);
    m_listRecentFiles.prepend(m_fileName);
    updateRecentFileActions();
    updateStatusBar();
    statusBar()->showMessage(tr("File loaded"), 2000);
  }
}     // MainWindow::loadFile()
```

Example (cont.)

Example (main-window.cpp, cont.)

```
void MainWindow::updateRecentFileActions()
{
  QMutableStringListIterator i(m_listRecentFiles);
  while (i.hasNext())
    if (!QFile::exists(i.next()))
      i.remove();
  //
```

Example (cont.)

Example (main-window.cpp, cont.)

```
for (int j = 0; j < enMaxRecentFiles; ++ j)
  if (j < m_listRecentFiles.count())
  {
    QString text = QString("&%1 %2")
      .arg(j + 1)
      .arg(strippedName(m_listRecentFiles[j]));
    m_apActionsRecent[j]->setText(text);
    m_apActionsRecent[j]->setData(m_listRecentFiles[j]);
    m_apActionsRecent[j]->setVisible(true);
  }
  else
    m_apActionsRecent[j]->setVisible(false);
```

Example (end)

Example (main-window.cpp, end)

```
  //
  m_pActionSeparator->setVisible(
    !m_listRecentFiles.empty());
}    // MainWindow::updateRecentFileActions()

void MainWindow::updateStatusBar()
{
  // ...
}
```

`QObject::tr()` Function

Declaration

```
static QString QObject::tr(
  const char *pcszSourceText,
  const char *pcszDisambiguation = 0,
  int n = -1);
```

Usage

⟨*context*⟩`::tr(`⟨*initial_text*⟩[`,` ⟨*comment_for_the_translator*⟩]`)`

Example of the Class Use

Example (my-window.h)

```
class MyWindow : public QWidget
{
  Q_OBJECT    // ⇒ ⟨context⟩ == MyWindow
  //
public:
  //
  MyWindow(QWidget *pParent = 0);
  // ...
```

Example of Translation Context Use

Example (my-window.cpp)

```
MyWindow::MyWindow(QWidget *pParent)
  : QWidget(pParent)
{
  QString str1 = tr("Letter");
  QString str2 = MyWindow::tr("Letter");     // the same
  QString str3 = MyForm::tr("Letter");
  QString str4 = MyPrintDialog::tr("Letter", "US paper size");
  // ...
}
```

Example of the Explicit Context Indication

Example (my-main.cpp)

```
int main(int nArgC, char *apszArgV[])
{
  QApplication app(nArgC, apszArgV);
  // ...
  QPushButton button(QObject::tr("Hello Qt!"));
  button.show();
  //
  return app.exec();
}
```

Example of Super-Explicit Context Indication

Example (my-main.cpp)

```
int main(int nArgC, char *apszArgV[])
{
  // ...
  QString str5 = QCoreApplication::translate(
    "Global Context", "Hello Qt!");
  // ...
}
```

Examples of tr() Wrong Use

Example

```
const char *pcszAppName = "Application Name";
QString translated = tr(pcszAppName);
```

Example

```
statusBar()->showMessage(tr("Page #: " + strPageNum));
```

Examples of `tr()` Wrong Use

Example

```
QString translated = tr("Application Name");
```

Example

```
statusBar()->showMessage(tr("Page #: %1").arg(nPageNum));
```

Translating String Arrays Inside Methods

Example (my-form.cpp)

```
void MyForm::MyForm(QWidget *pParent)
  : QWidget(pParent)
{
  static const char *s_apcszItems[] =
  {
    QT_TR_NOOP("Red"),
    QT_TR_NOOP("Green"),
    QT_TR_NOOP("Blue"),
    0
  };
  for (int i = 0; s_apcszItems[i]; ++i)
    m_pComboBox->addItem(tr(s_apcszItems[i]));
  // ...
```

Translating Global String Arrays

Example (my-form.cpp)

```
static const char *g_apcszItems[] =
{
  QT_TRANSLATE_NOOP("MyForm", "Red"),
  QT_TRANSLATE_NOOP("MyForm", "Green"),
  QT_TRANSLATE_NOOP("MyForm", "Blue"),
  0
};

void MyForm::MyForm(QWidget *pParent)
  : QWidget(pParent)
{
  // ...
```

Preventing Skipped Translations

Example (CMakeLists.txt)

```
add_definitions(-DQT_NO_CAST_FROM_ASCII)
```

Example (my-form.cpp)

```
void MyForm::MyForm(QWidget *pParent)
  : QWidget(pParent)
{
  QString str1 = tr("One");
  QString str2 = QLatin1String("Two-original");
  QString str3 = "Three-original";     // Compilation error
  // ...
```

Creating Translation Files

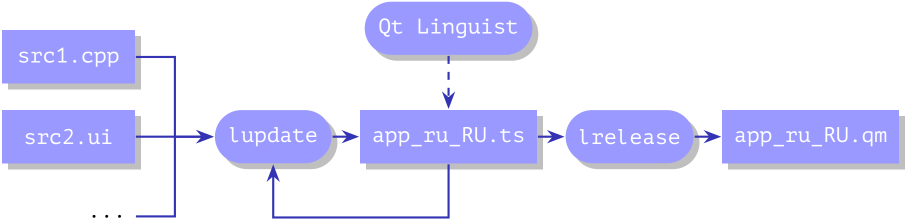
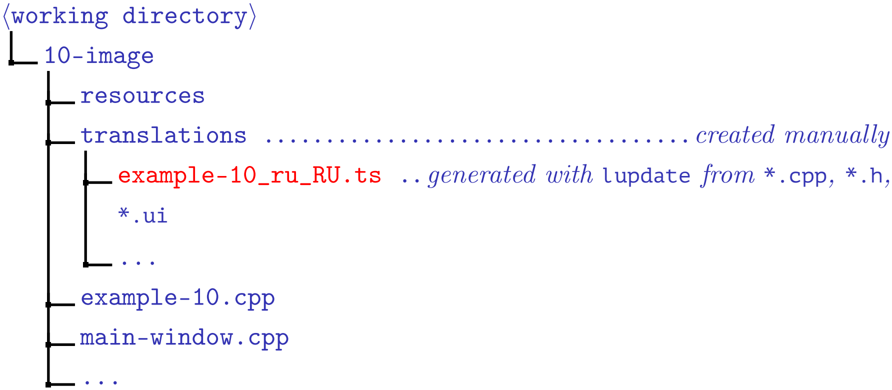


Figure 5: creating localization files with `lupdate` and `lrelease` tools

Project Structure



```
⟨working directory⟩
└─ 10-image
   ├─ resources
   ├─ translations ................................ created manually
   │  ├─ example-10_ru_RU.ts .. generated with lupdate from *.cpp, *.h,
   │  │  *.ui
   │  └─ ...
   ├─ example-10.cpp
   ├─ main-window.cpp
   └─ ...
```

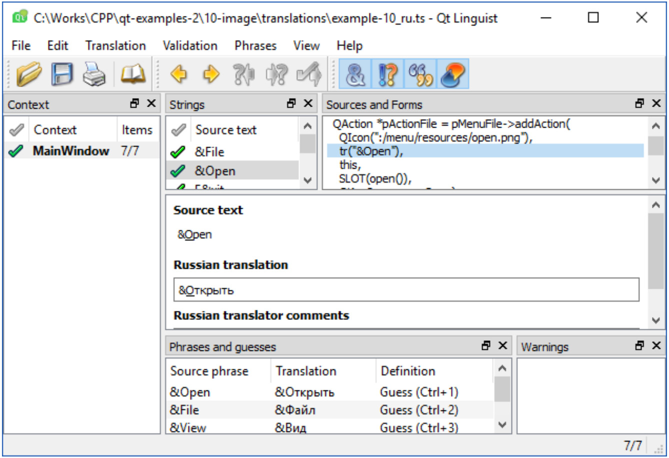
Figure 6: a structure for the project directory using translation source files (*.ts)

Example of a Translation Source File Contents

Example (translations\example-10_ru_RU.ts)

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE TS>
<TS version="2.1" language="ru_RU">
<context>
    <name>MainWindow</name>
    <message>
        <location filename="../main-window.cpp" line="29"/>
        <location filename="../main-window.cpp" line="54"/>
        <source>&amp;File</source>
        <translation type="unfinished"></translation>
    </message>
```

Editing a Translation



Context	Items
✓ MainWindow	7/7

Source text
✓ &File
✓ &Open
✓ E&xit

```
QAction *pActionFile = pMenuFile->addAction(
  QIcon(":/menu/resources/open.png"),
  tr("&Open"),
  this,
  SLOT(open()),
```

Source text

&Open

Russian translation

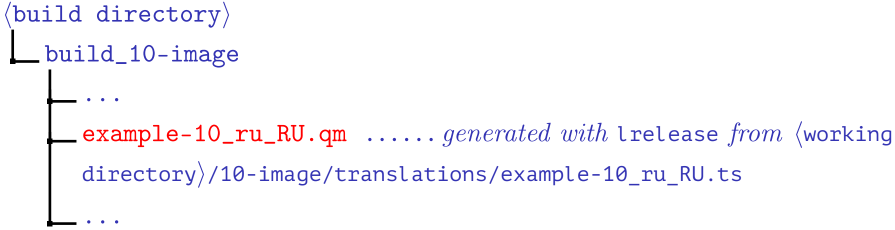
&Открыть

Russian translator comments

Source phrase	Translation	Definition
&Open	&Открыть	Guess (Ctrl+1)
&File	&Файл	Guess (Ctrl+2)
&View	&Вид	Guess (Ctrl+3)

Figure 7: a window of Qt Linguist translation editor

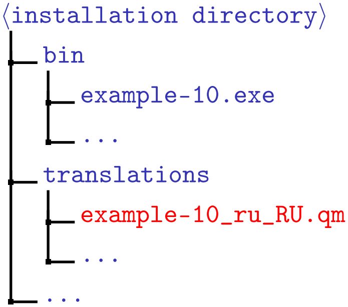
Project Structure (end)



```
⟨build directory⟩
└─ build_10-image
   ├─ ...
   ├─ example-10_ru_RU.qm ...... generated with lrelease from ⟨working
   │  directory⟩/10-image/translations/example-10_ru_RU.ts
   └─ ...
```

Figure 8: a structure for the project build directory

Structure for the Installation Directory



⟨installation directory⟩
- bin
  - example-10.exe
  - ...
- translations
  - example-10_ru_RU.qm
  - ...
- ...

Figure 9: a structure for the application install directory

Example

Example (example-10.cpp)

```
#include "main-window.h"

#include <QApplication>
#include <QScopedPointer>
#include <QMessageBox>
#include <QTranslator>

int main(int nArgC, char *apszArgV[])
{
  QApplication app(nArgC, apszArgV);
  //
  bool bTranslationLoaded = false;
  QTranslator translator;
```

Example (cont.)

Example (example-10.cpp, cont.)

```
const char *aszTranslationDirs[] =
{
  ".", "./10-image", "../translations", ":/translations", 0
};
for (
  const char **ppszDir = aszTranslationDirs;
  *ppszDir != 0; ++ ppszDir)
{
  bTranslationLoaded = translator.load(
    QLocale::system(), "example-10", "_", *ppszDir);
  if (bTranslationLoaded)
    break;
}
```

Example (cont.)

Example (example-10.cpp, end)

```
  if (bTranslationLoaded)
    app.installTranslator(&translator);
  else
    QMessageBox::warning(0, "Translation", "Not loaded");
  //
  QScopedPointer <MainWindow> mainWindow(
    new MainWindow);
  mainWindow->show();
  //
  return app.exec();
}    // main()
```

Example (cont.)

Example (CMakeLists.txt)

```
cmake_minimum_required(VERSION 2.8.11)

project(10-image)

find_package(Qt5 REQUIRED Widgets LinguistTools)

# Translation

set(
  FILES_TO_TRANSLATE
  example-10.cpp
  main-window.cpp)
```

Example (cont.)

Example (CMakeLists.txt)

```
set(
  TS_FILES
  translations/example-10_ru_RU.ts)

option(
  UPDATE_TRANSLATIONS
  "Update source translation (files translations/*.ts)."
  ON)
```

Example (cont.)

Example (CMakeLists.txt, cont.)

```
if(UPDATE_TRANSLATIONS)
  message(
    STATUS "UPDATE_TRANSLATIONS option is set.")
  qt5_create_translation(
    QM_FILES ${FILES_TO_TRANSLATE} ${TS_FILES})
else()
  qt5_add_translation(
    QM_FILES ${TS_FILES})
endif()
```

Example (cont.)

Example (CMakeLists.txt, cont.)

```
add_custom_target(
  translations
  DEPENDS ${QM_FILES})

set_property(
  DIRECTORY
  PROPERTY CLEAN_NO_CUSTOM TRUE)
```

Example (CMakeLists.txt, cont.)

```
# Main executable

qt5_wrap_cpp(
  MOC_WRAPPERS
  main-window.h)

qt5_add_resources(
  QRC_WRAPPERS
  example-10.qrc
  OPTIONS -no-compress)
```

Example (cont.)

Example (CMakeLists.txt, cont.)

```
add_executable(
  example-10 WIN32
  ${FILES_TO_TRANSLATE}
  ${MOC_WRAPPERS}
  ${QRC_WRAPPERS})
```

Example (CMakeLists.txt, end)

```
target_link_libraries(
  example-10 Qt5::Widgets)


# ...
```

Example (end)

Example (build.cmd)

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

cmake^
  -G "MinGW Makefiles"^
  -D CMAKE_PREFIX_PATH="C:\Qt\Qt5.7.0\5.7\mingw53_32"^
  -D UPDATE_TRANSLATIONS=ON^
  D:\Work\10-image
rem  -D UPDATE_TRANSLATIONS=OFF^

mingw32-make
mingw32-make translations
rem linguist
mingw32-make translations
```

Example

Example (CMakeLists.txt)

```
# ...
# Installation

install(
  TARGETS example-10
  RUNTIME DESTINATION bin)

install(
  FILES ${QM_FILES}
  DESTINATION translations)
```

Example (cont.)

Example (CMakeLists.txt, cont.)

```
get_property(
  LIB_QT5CORE_PATH
  TARGET Qt5::Core
  PROPERTY IMPORTED_LOCATION_RELEASE)

get_filename_component(
  QT_LIBRARY_DIR "${LIB_QT5CORE_PATH}" DIRECTORY)

set(
  EXE_PATH bin/example-10${CMAKE_EXECUTABLE_SUFFIX})
```

Example (cont.)

Example (CMakeLists.txt, end)

```
include(DeployQt4)

install_qt4_executable(
  "${EXE_PATH}"          # executable file
  ""                     # modules
  ""                     # libraries
  "${QT_LIBRARY_DIR}")   # directories for libraries search
```

Example (end)

Example (build.cmd)

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

cmake^
  -G "MinGW Makefiles"^
  -D CMAKE_PREFIX_PATH="C:\Qt\Qt5.7.0\5.7\mingw53_32"^
  -D CMAKE_INSTALL_PREFIX=D:\install\10-image^
  -D UPDATE_TRANSLATIONS=ON^
  D:\Work\10-image
```