

# Как создать новый проект с нуля, используя исходники проекта HelloQt из SDK

Рекомендуется использовать среду **Visual Studio 2019**, **toolset v142** и **Qt 5.15.1**.

## 1. Установите Qt VS Tools

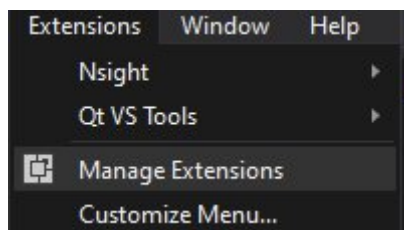


Рис. 1

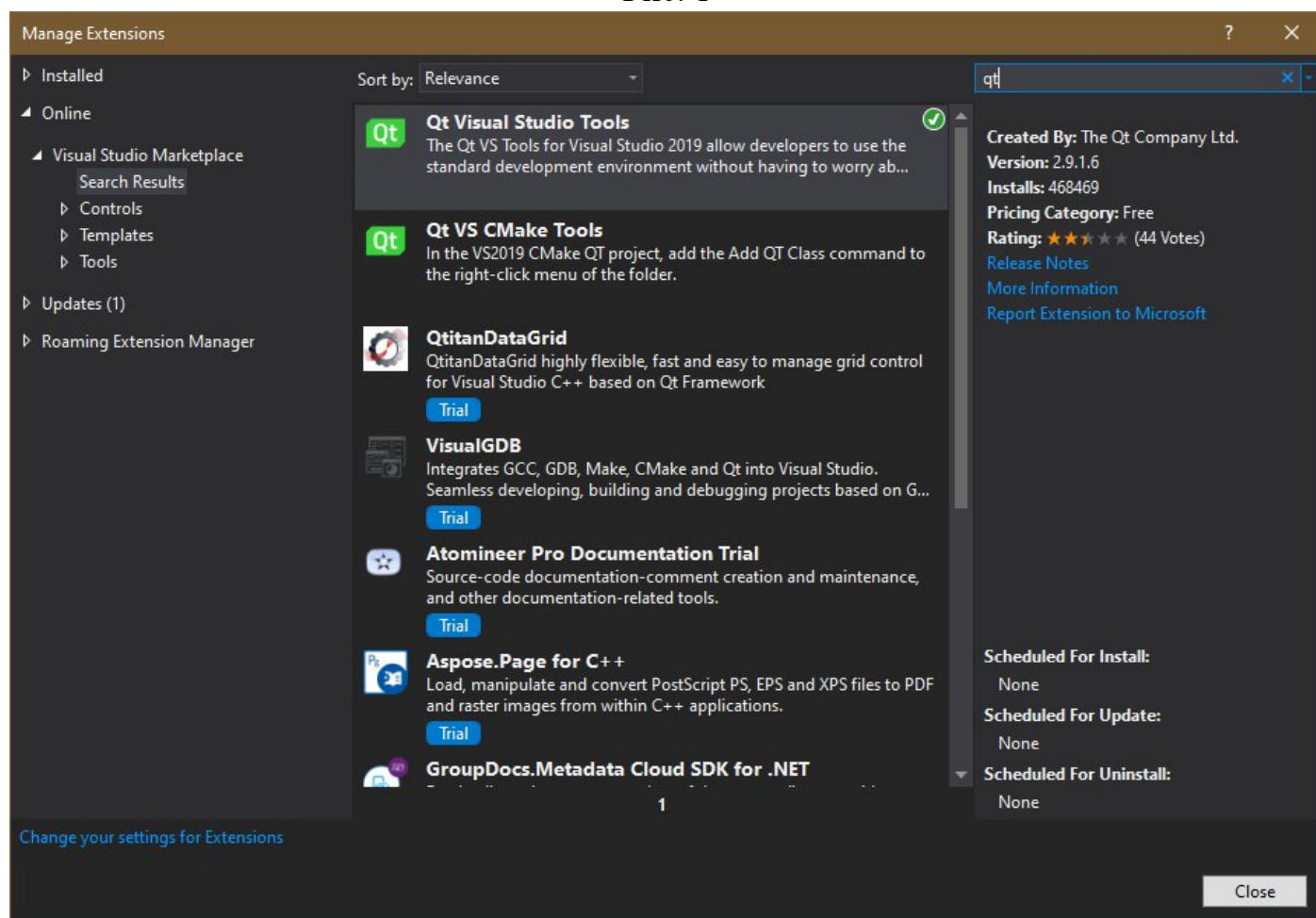


Рис. 2

## 2. Добавьте переменные среды:

QtToolsPath = C:\Qt\5.15.1\msvc2019\_64\bin

QtRoot = C:\Qt\5.15.1\msvc2019\_64

NCadSDK = C:\Program Files\Nanosoft\nanoCAD x64 22.0\SDK

Нажмите клавиши **Win+R** на клавиатуре, введите **sysdm.cpl** и нажмите Enter.

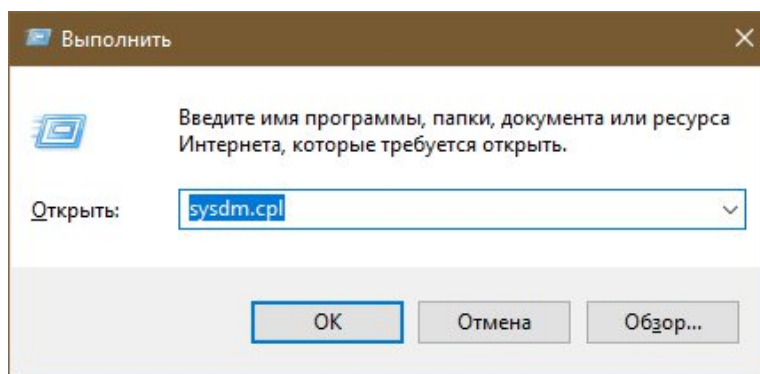


Рис. 3

На вкладке «Дополнительно» нажмите кнопку «Переменные среды...»

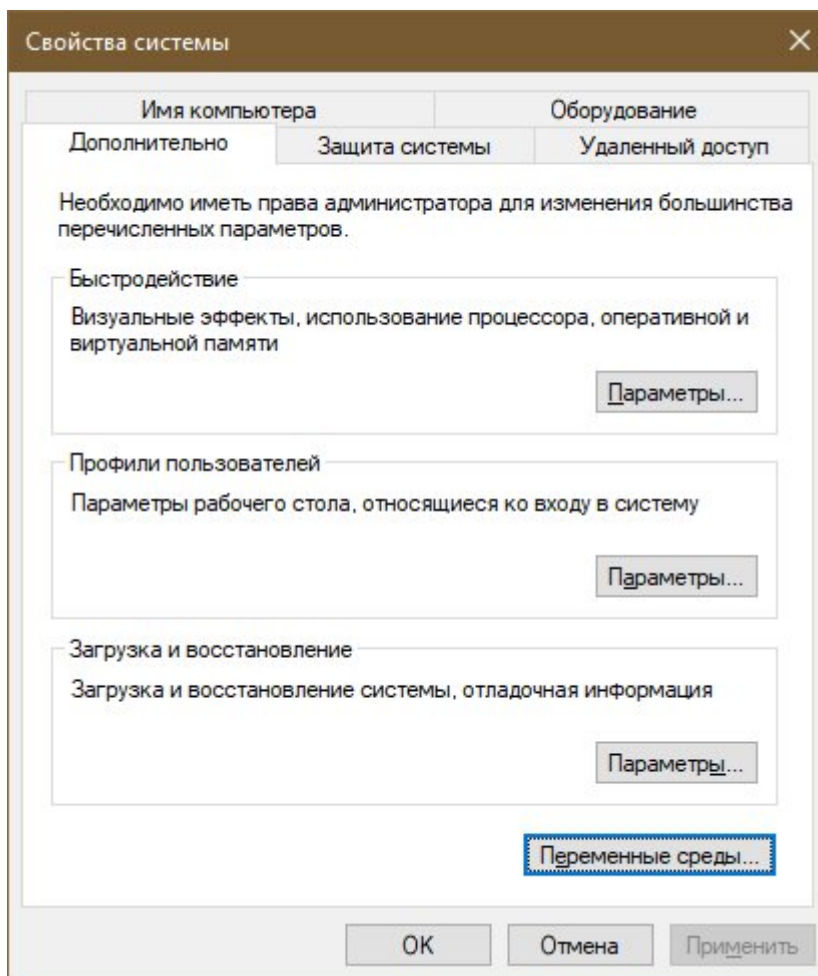


Рис. 4

В разделе «Переменные среды пользователя» (если требуется изменение только для текущего пользователя) или «Системные переменные» нажмите кнопку «Создать...»

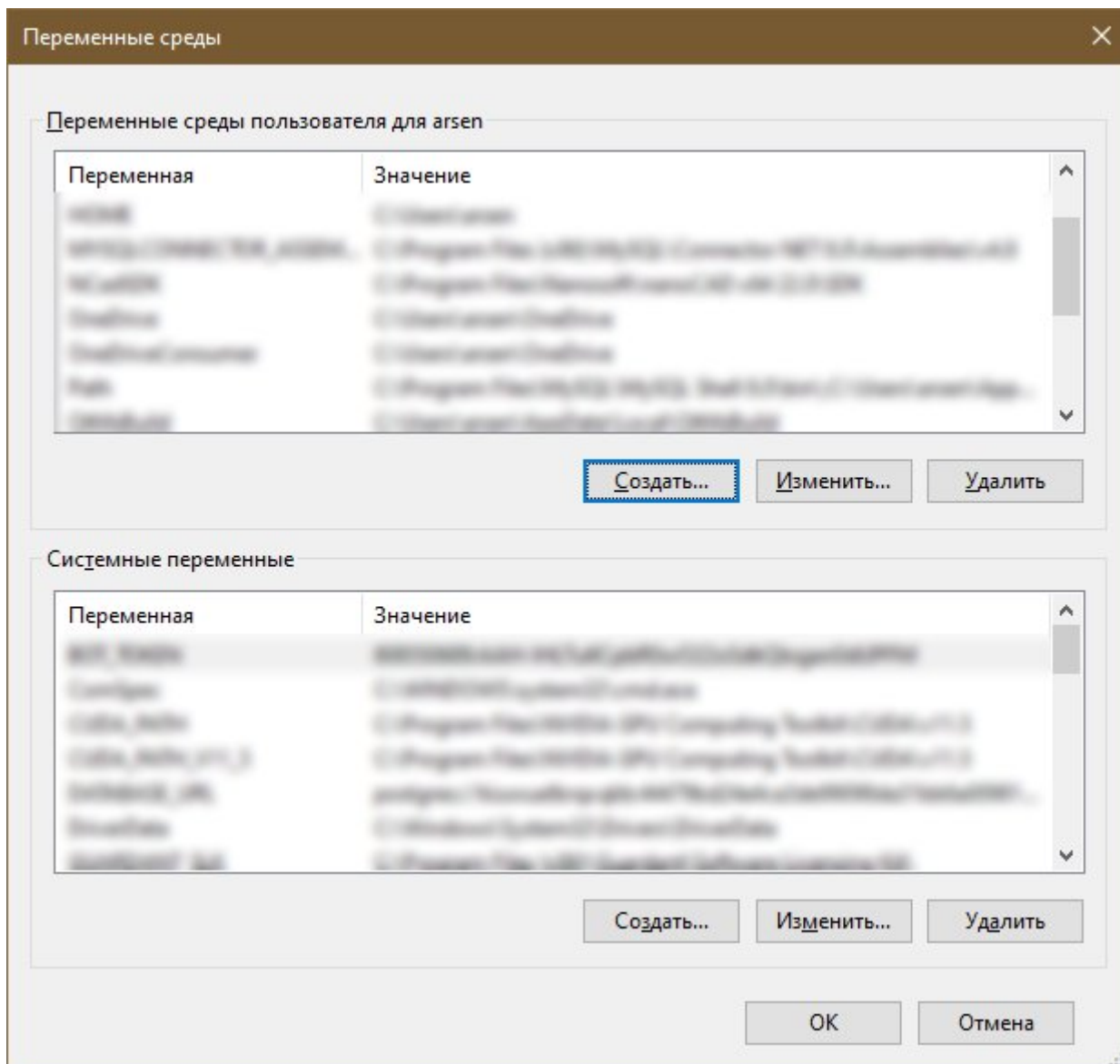


Рис. 5

Введите в поле «Имя переменной:» QtToolsPath, а в «Значение переменной:» путь к папке C:\Qt\5.15.1\msvc2019\_64\bin. Если у вас Qt лежит в другом месте или установлена другая версия, скорректируйте путь.

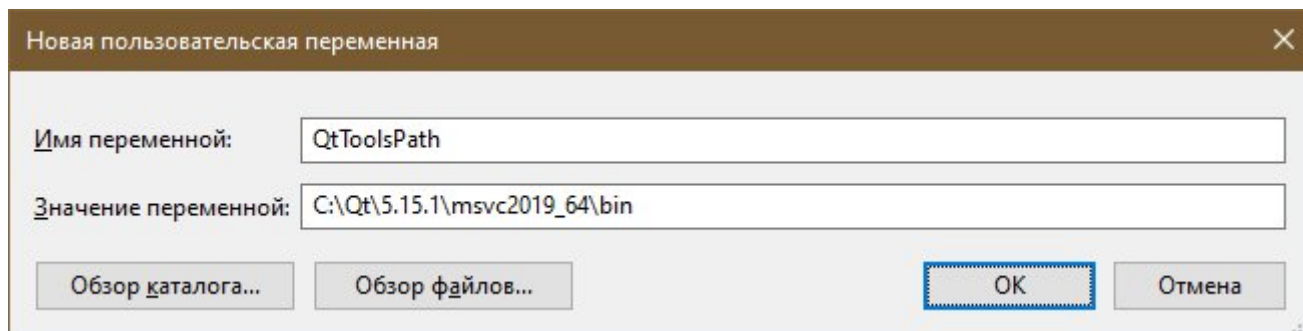


Рис. 6

Аналогично для переменной QtRoot = C:\Qt\5.15.1\msvc2019\_64

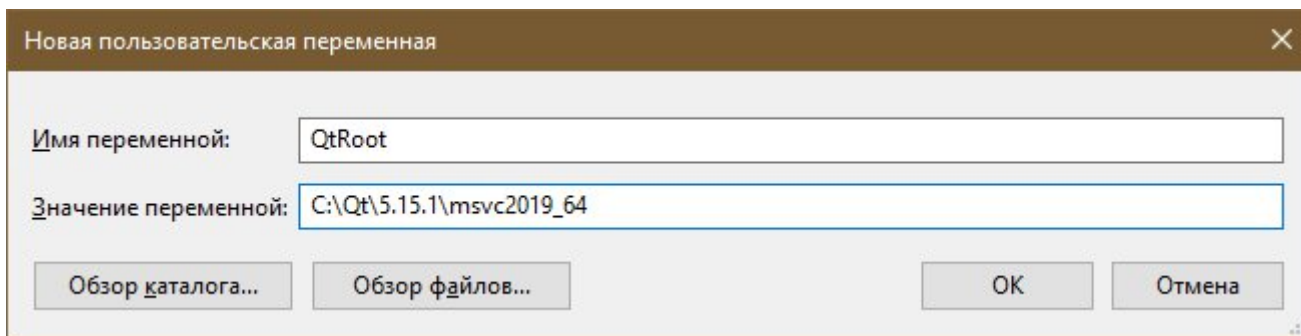


Рис. 7

И последняя переменная NCadSDK должна указывать на путь до папки NanoCad SDK (например, C:\Program Files\Nanosoft\nanoCAD x64 22.0\SDK)

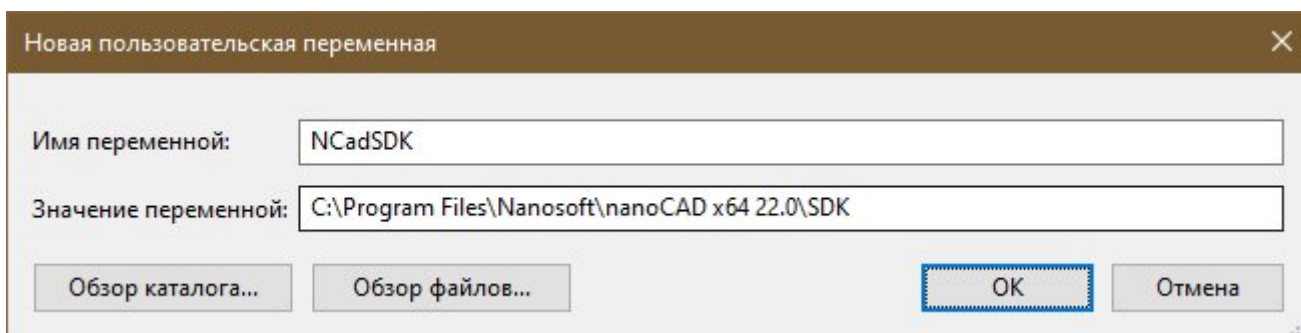


Рис. 8

3. Добавьте версию Qt в расширение Qt Vs Tools

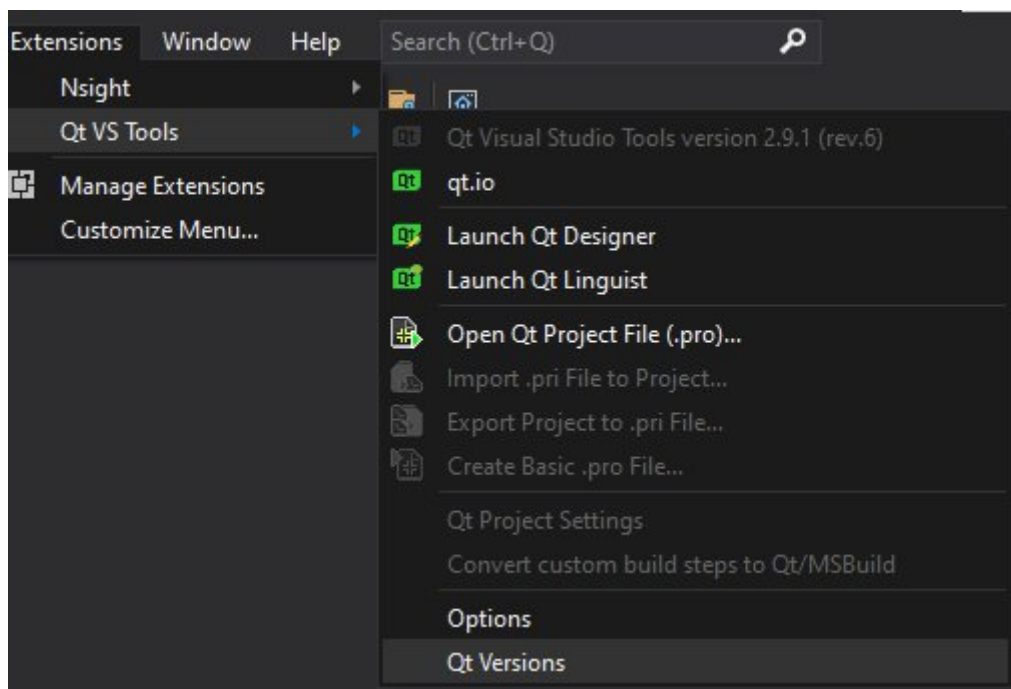


Рис. 9

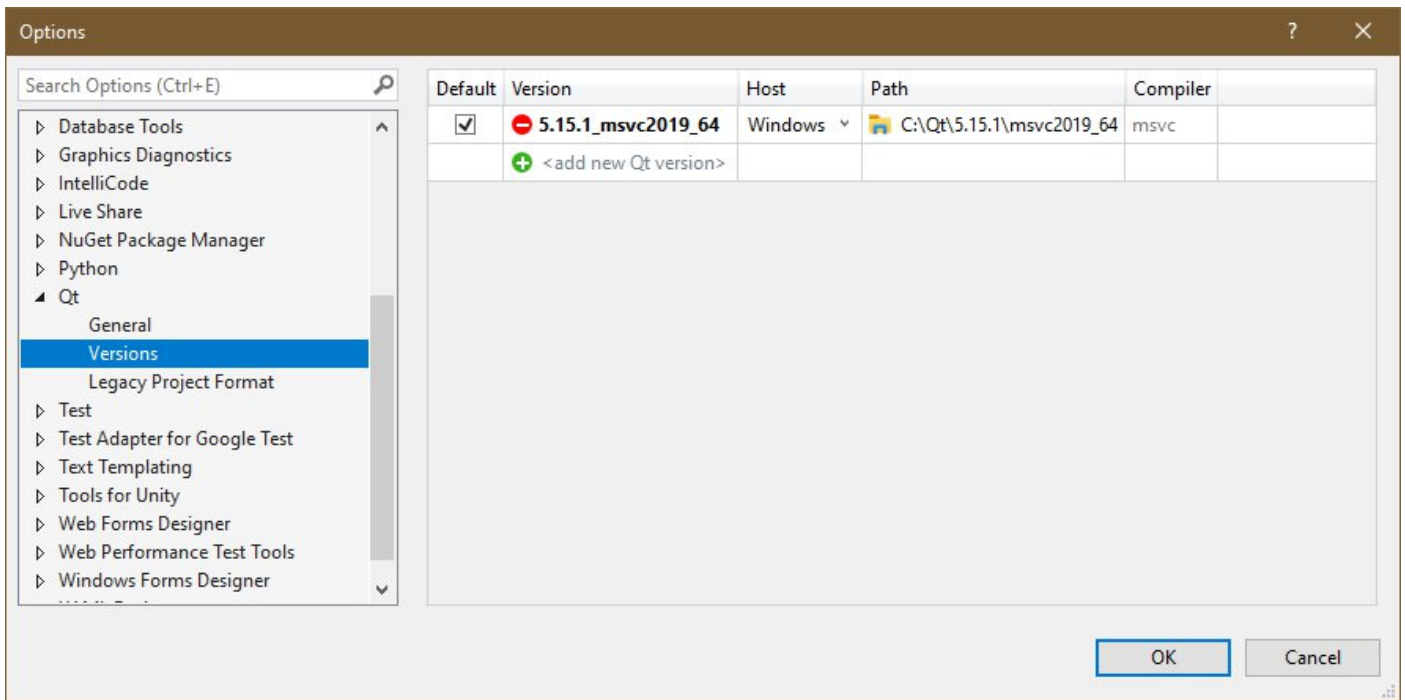


Рис. 10

#### 4. Создайте новый проект Qt Class Library

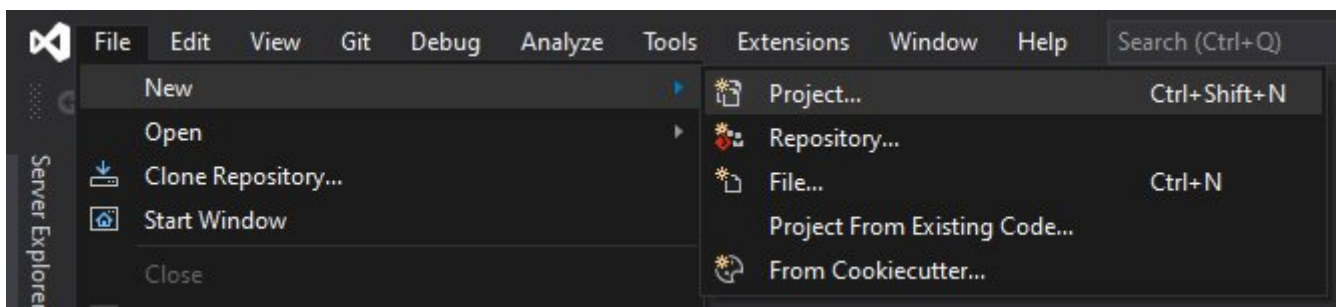


Рис. 11

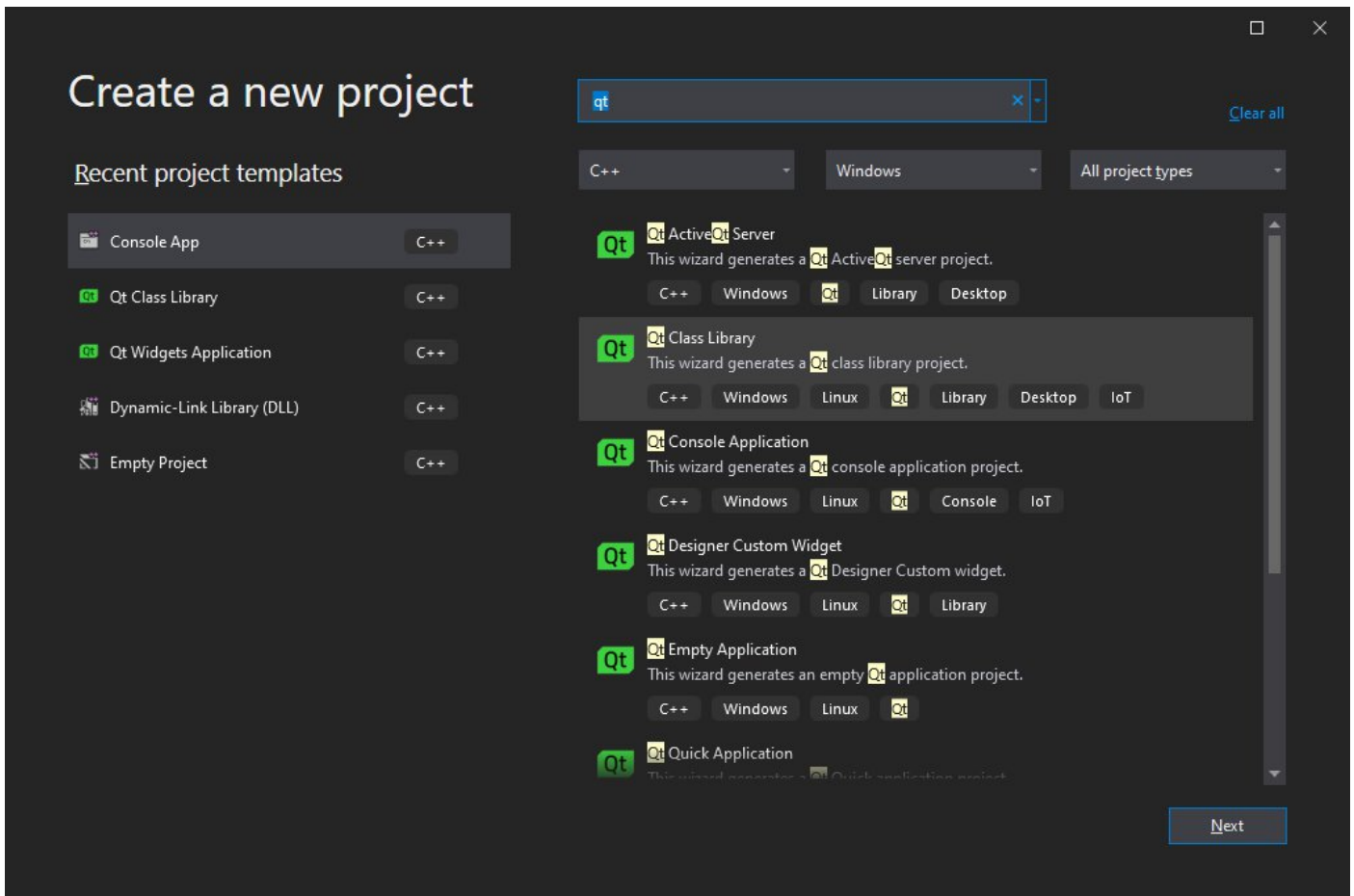


Рис. 12

Укажите имя проекта и папку для его расположения.

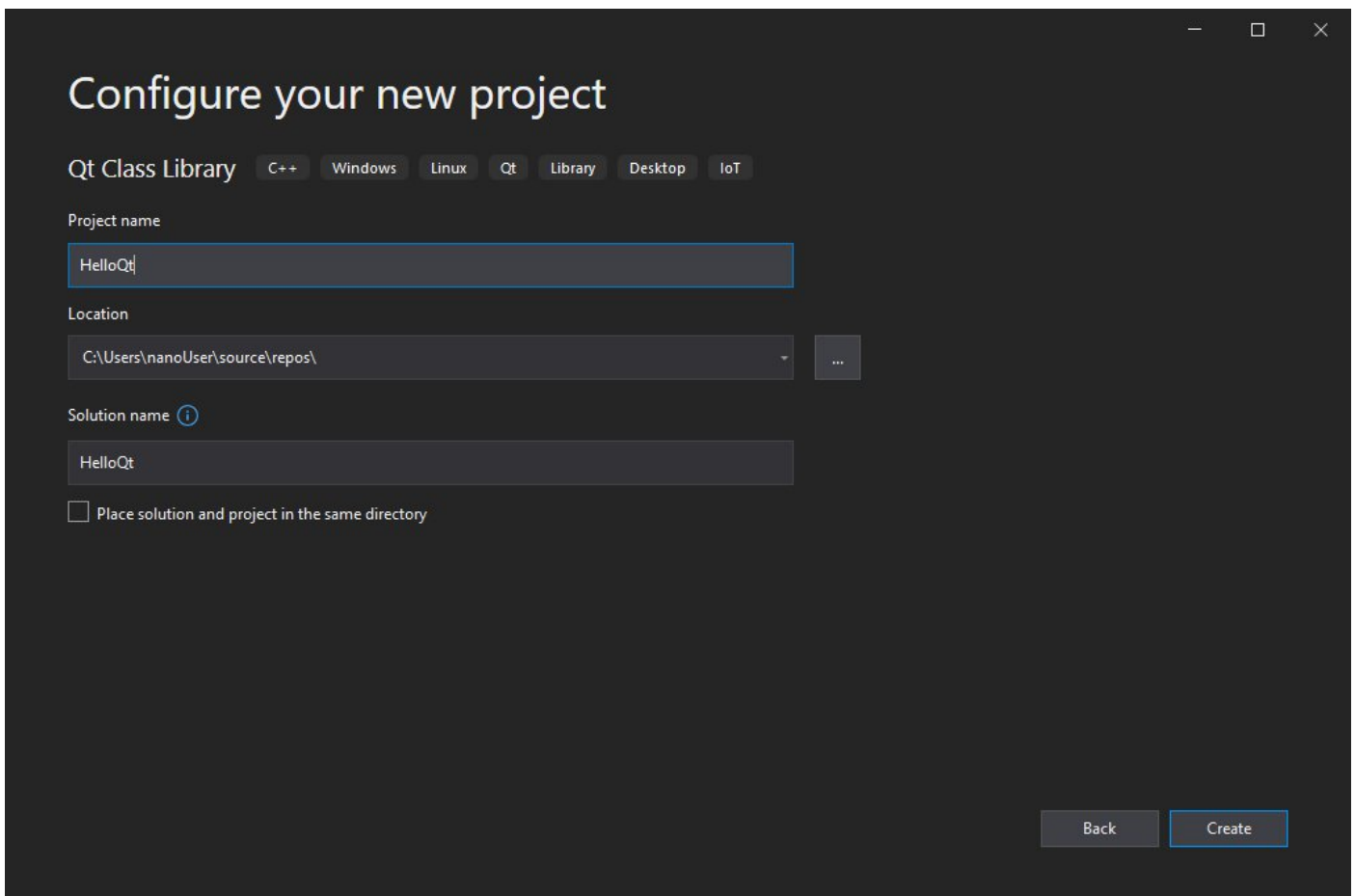


Рис. 13

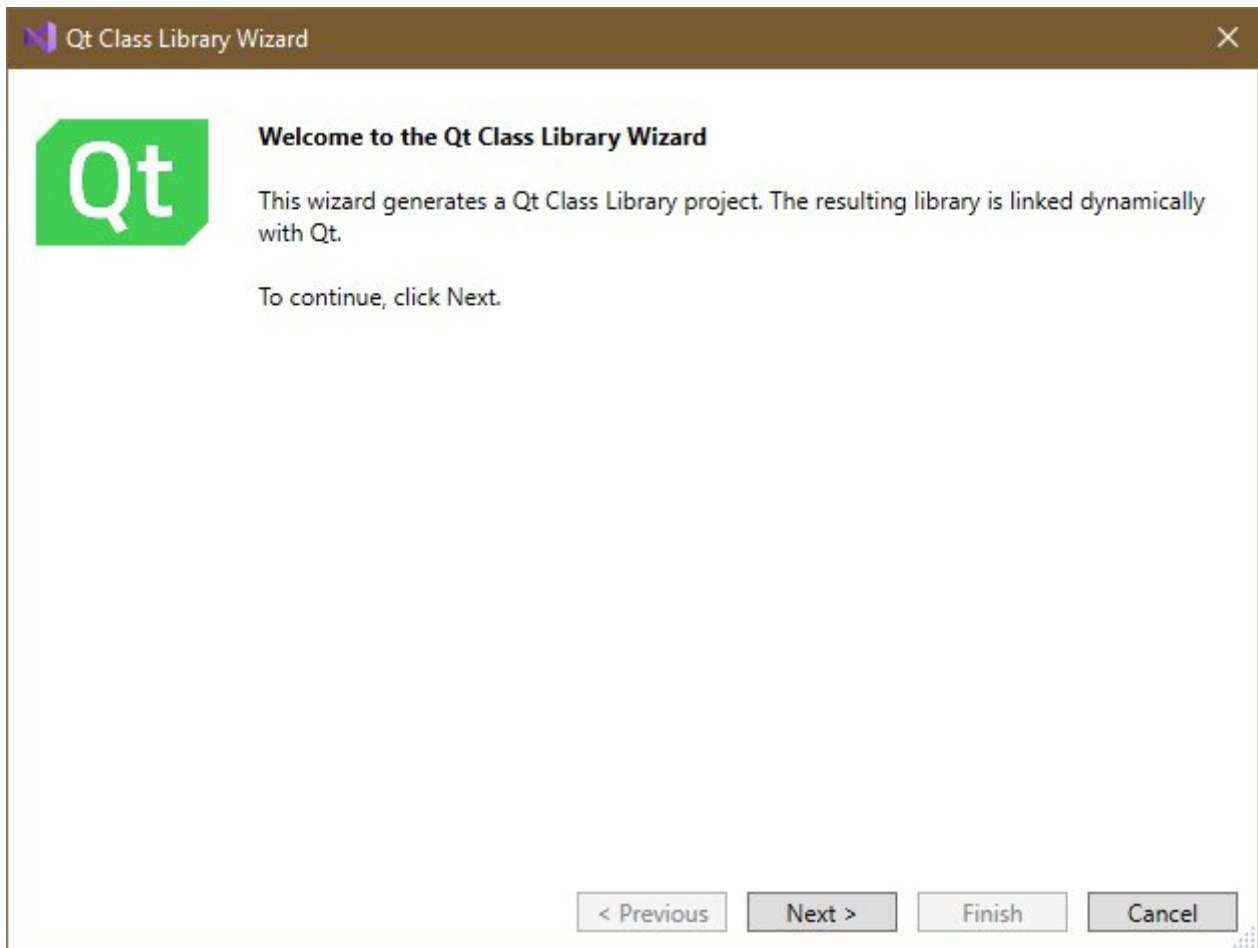


Рис. 14

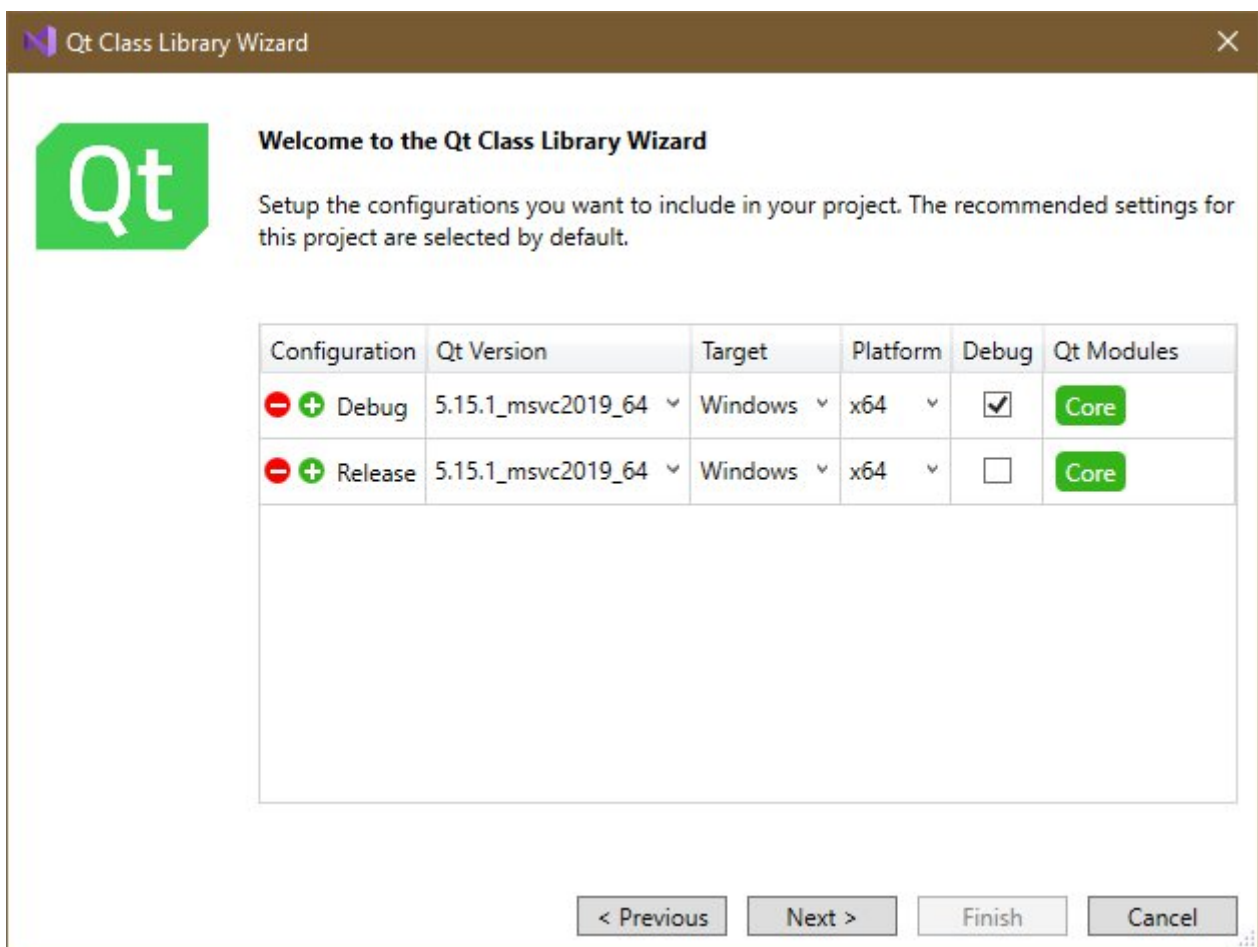


Рис.15

Поставьте галочку Precompiled header

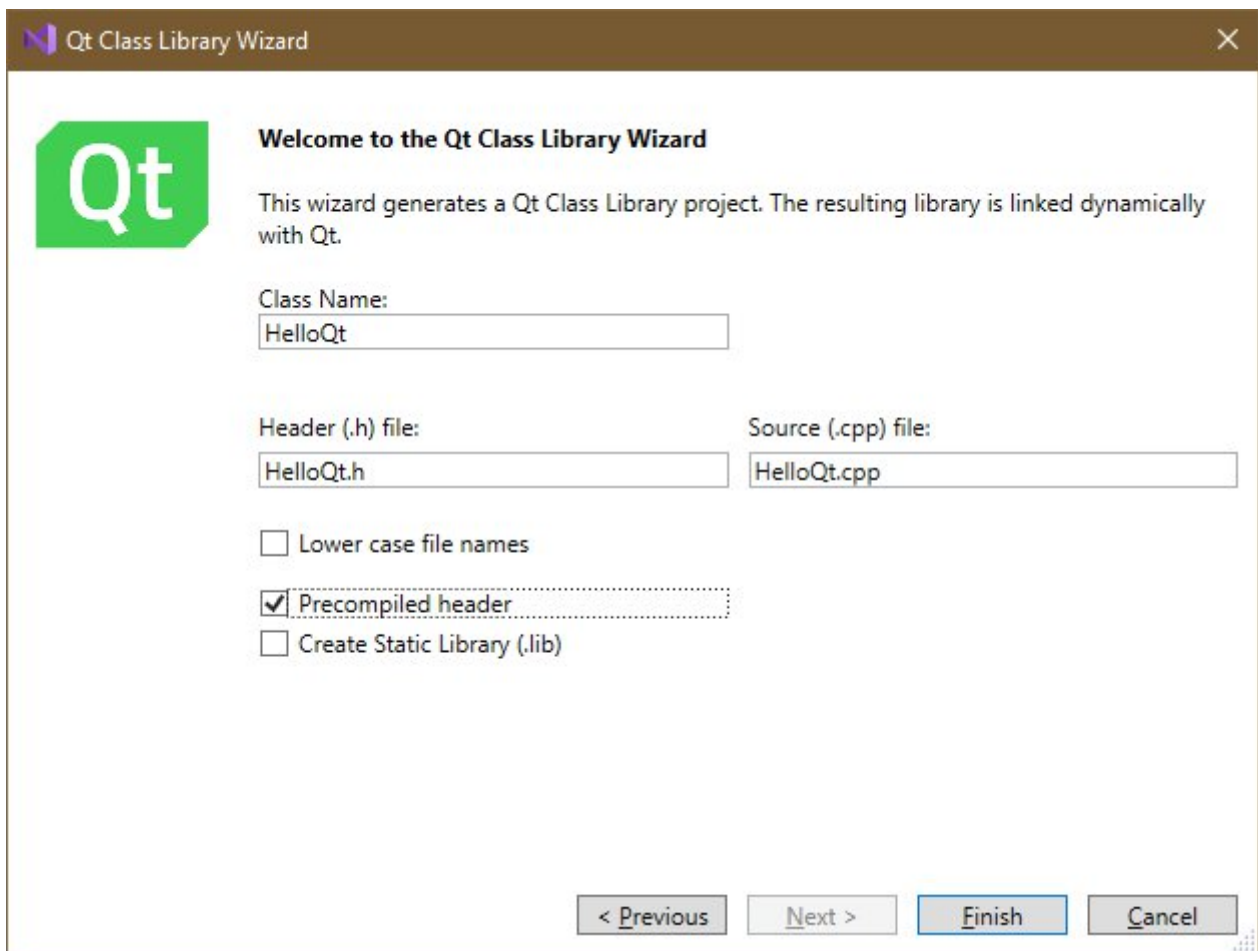


Рис. 16

В результате будет создан проект, состоящий из нескольких файлов (в зависимости от версии Visual Studio список файлов может отличаться):

- **HelloQt.cpp**
- HelloQt.h
- helloqt\_global.h
- **stdafx.cpp**
- **stdafx.h**



5. Для **каждой** созданной конфигурации в менеджере свойств (Menu -> View -> **Other Windows -> Property Manager**) подключите страницы свойств `rxsdk_debugcfg.props` и `rxsdk_releasecfg.props` из директории `$(NanoCadSDK)\include\arxgate` соответственно:

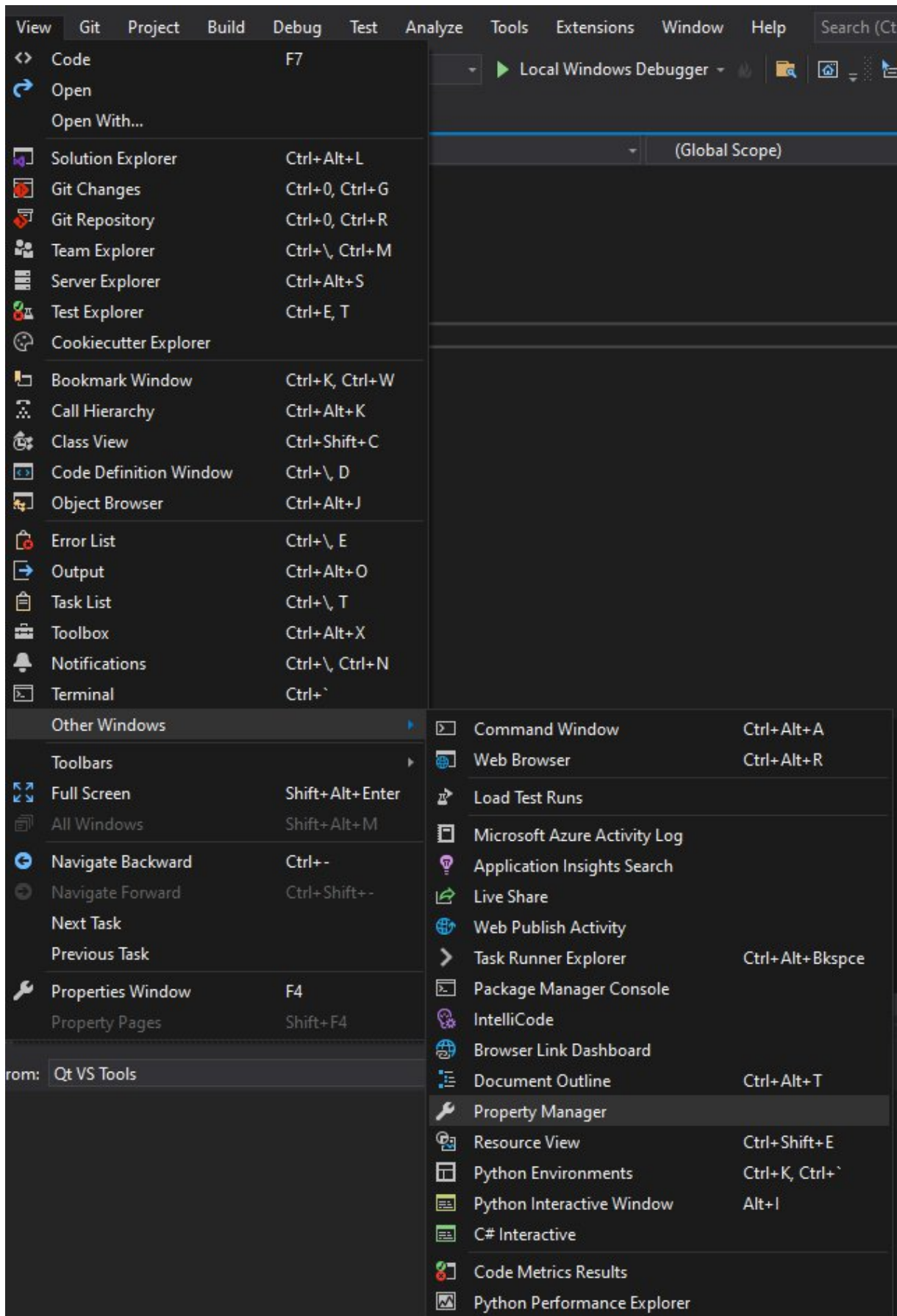


Рис. 17

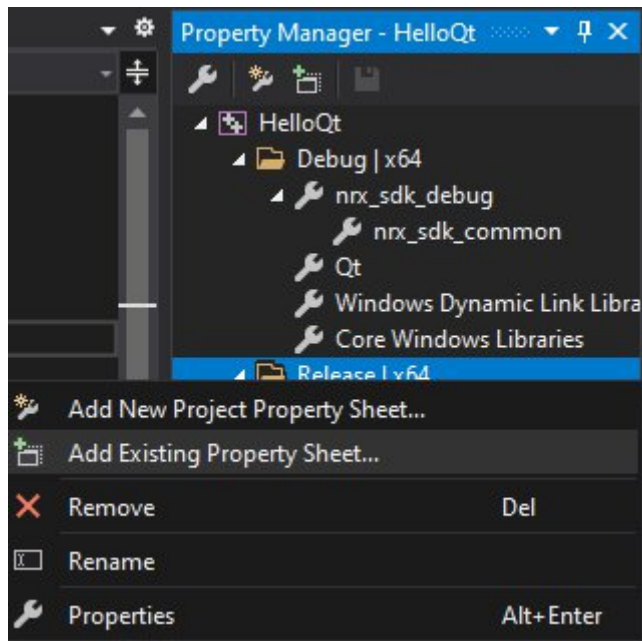


Рис. 18

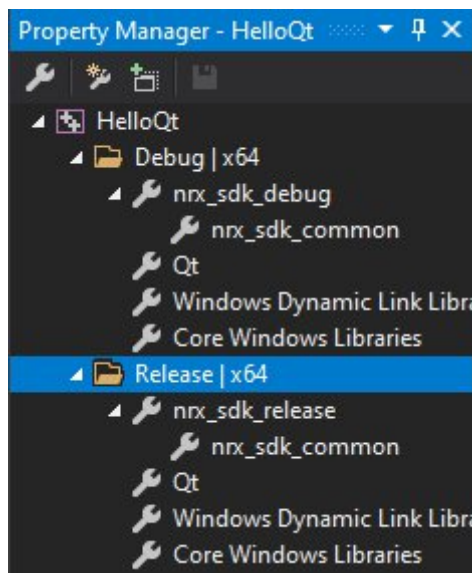


Рис. 19

Чтобы проект можно было свободно перемещать, измените в файле проекта (\*.vcxproj) относительный путь к страницам свойств на абсолютный следующим образом:

Было:

```

52 <ImportGroup Label="PropertySheets" Condition="'$(Configuration)|$(Platform)' == 'Debug|x64'"
53 <Import Project="$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props" Condition="exists('$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props')" />
54 <Import Project="$(QtMsBuild)\Qt.props" />
55 <Import Project="..\..\..\..\..\Program Files\Nanosoft\nanoCAD x64 22.0\SDK\include\arxgate\rxsdk_debugcfg.props" />
56 </ImportGroup>
57 <ImportGroup Label="PropertySheets" Condition="'$(Configuration)|$(Platform)' == 'Release|x64'"
58 <Import Project="$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props" Condition="exists('$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props')" />
59 <Import Project="$(QtMsBuild)\Qt.props" />
60 <Import Project="..\..\..\..\..\Program Files\Nanosoft\nanoCAD x64 22.0\SDK\include\arxgate\rxsdk_releasecfg.props" />
61 </ImportGroup>

```

Рис. 20

```

<ImportGroup Label="PropertySheets" Condition="'$(Configuration)|$(Platform)' ==
'Debug|x64'">
  <Import Project="$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props"
Condition="exists('$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props')">

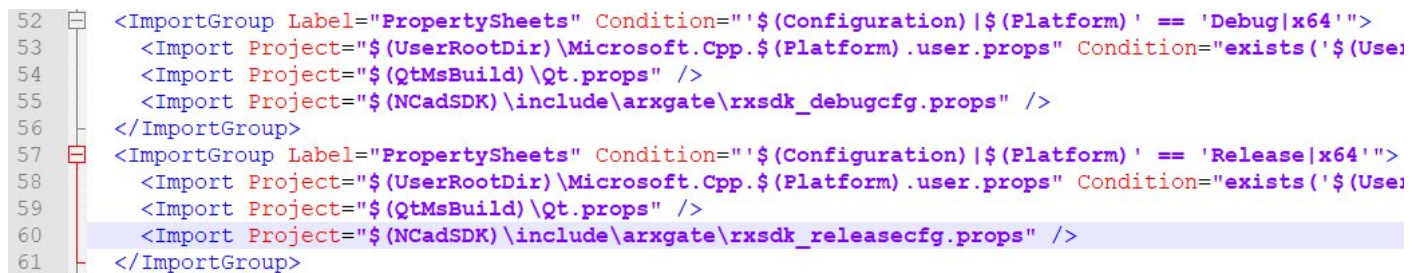
```

```

Label="LocalAppDataPlatform" />
  <Import Project="$(QtMsBuild)\Qt.props" />
  <Import Project="..\..\..\..\Program Files\Nanosoft\nanoCAD x64
22.0\SDK\include\arxgate\rxsdk_debugcfg.props" />
</ImportGroup>
<ImportGroup Label="PropertySheets" Condition="'$(Configuration)|$(Platform)' ==
'Release|x64'">
  <Import Project="$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props"
Condition="exists('$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props')">
Label="LocalAppDataPlatform" />
  <Import Project="$(QtMsBuild)\Qt.props" />
  <Import Project="..\..\..\..\Program Files\Nanosoft\nanoCAD x64
22.0\SDK\include\arxgate\rxsdk_releasecfg.props" />
</ImportGroup>

```

Стало:



```

52 <ImportGroup Label="PropertySheets" Condition="'$(Configuration)|$(Platform)' == 'Debug|x64'">
53   <Import Project="$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props" Condition="exists('$(User
54   <Import Project="$(QtMsBuild)\Qt.props" />
55   <Import Project="$(NCadSDK)\include\arxgate\rxsdk_debugcfg.props" />
56 </ImportGroup>
57 <ImportGroup Label="PropertySheets" Condition="'$(Configuration)|$(Platform)' == 'Release|x64'">
58   <Import Project="$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props" Condition="exists('$(User
59   <Import Project="$(QtMsBuild)\Qt.props" />
60   <Import Project="$(NCadSDK)\include\arxgate\rxsdk_releasecfg.props" />
61 </ImportGroup>

```

Рис. 21

```

<ImportGroup Label="PropertySheets" Condition="'$(Configuration)|$(Platform)' ==
'Debug|x64'">
  <Import Project="$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props"
Condition="exists('$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props')">
Label="LocalAppDataPlatform" />
  <Import Project="$(QtMsBuild)\Qt.props" />
  <Import Project="$(NCadSDK)\include\arxgate\rxsdk_debugcfg.props" />
</ImportGroup>
<ImportGroup Label="PropertySheets" Condition="'$(Configuration)|$(Platform)' ==
'Release|x64'">
  <Import Project="$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props"
Condition="exists('$(UserRootDir)\Microsoft.Cpp.$(Platform).user.props')">
Label="LocalAppDataPlatform" />
  <Import Project="$(QtMsBuild)\Qt.props" />
  <Import Project="$(NCadSDK)\include\arxgate\rxsdk_releasecfg.props" />
</ImportGroup>

```

Сохраните файл и вернувшись в Visual Studio нажмите кнопку «Reload All»

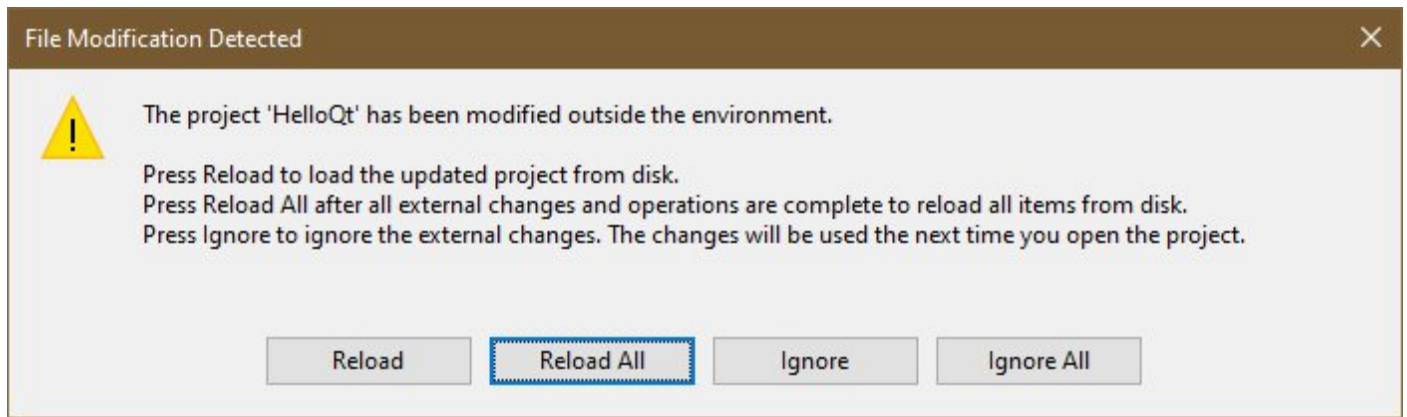


Рис. 22

6. Добавьте пути к заголовочным файлам

Menu -> Project -> Properties -> Configuration Properties -> C/C++ -> General -> Additional Include Directories

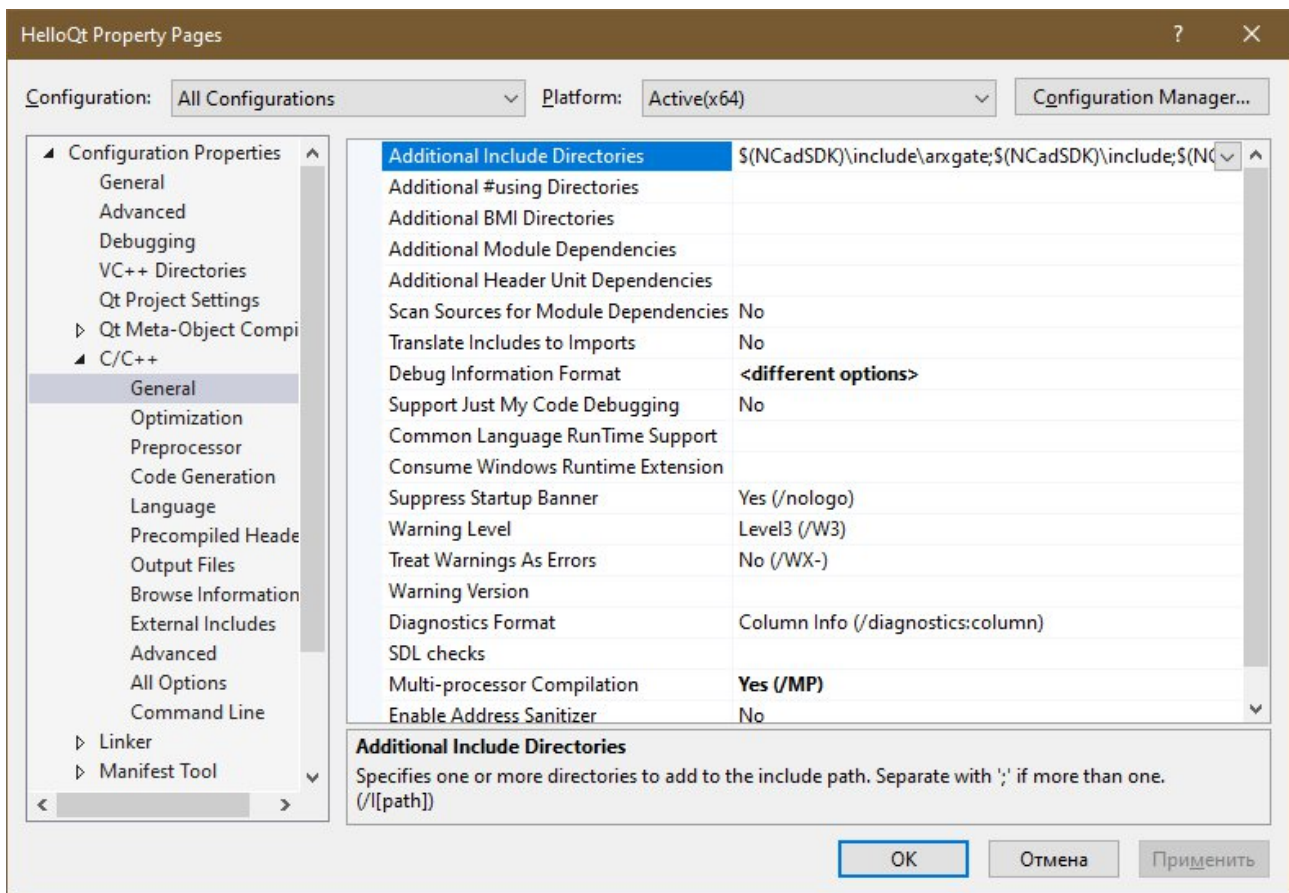


Рис. 23

```
$(QtRoot)\include\;$(QtRoot)\include\QtCore;$(QtRoot)\include\QtGui;$(QtRoot)\include\QtWidgets;$(QtRoot)\include\QtWinExtras;$(QtRoot)\include\QtXml;$(QtRoot)\include\QtANGLE;
```

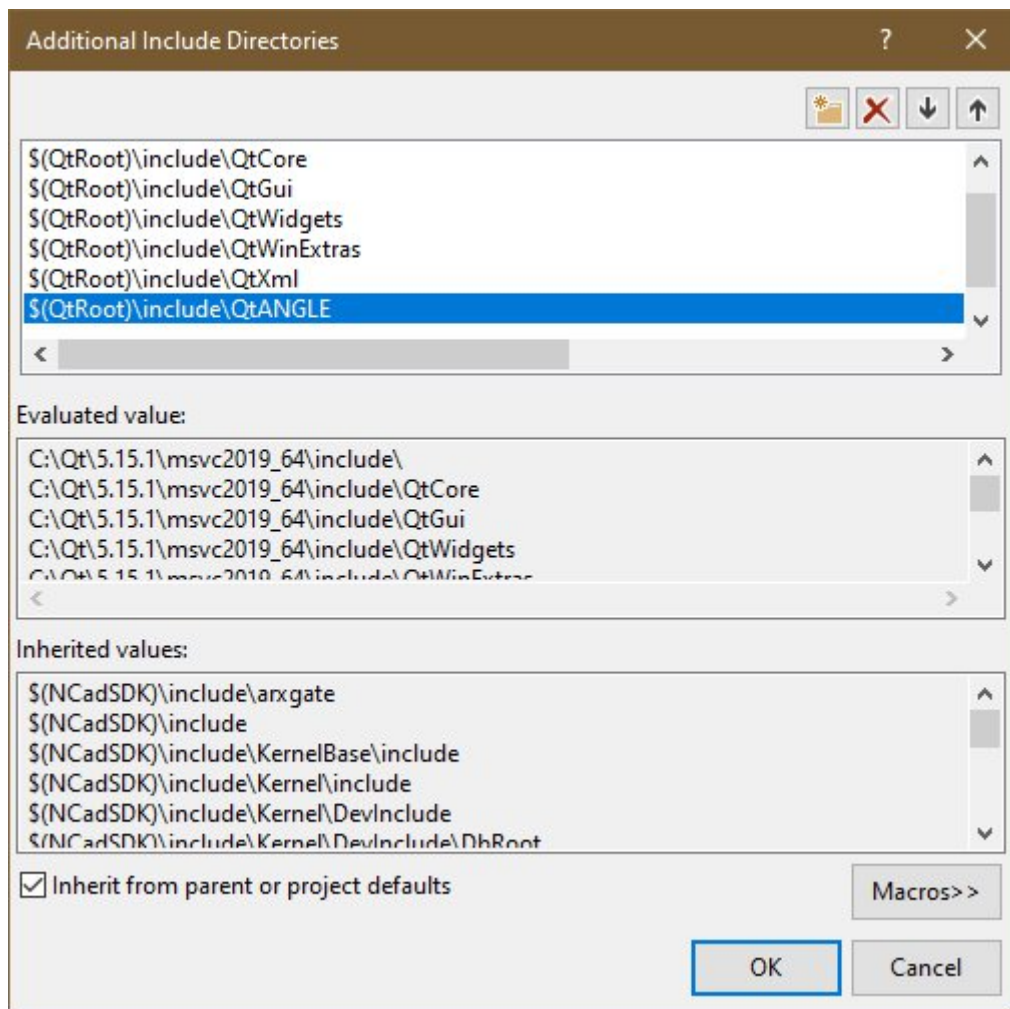


Рис. 24

7. Добавьте путь к библиотекам Qt

Menu -> Project -> Properties -> Configuration Properties -> Linker -> General -> Additional Library Directories

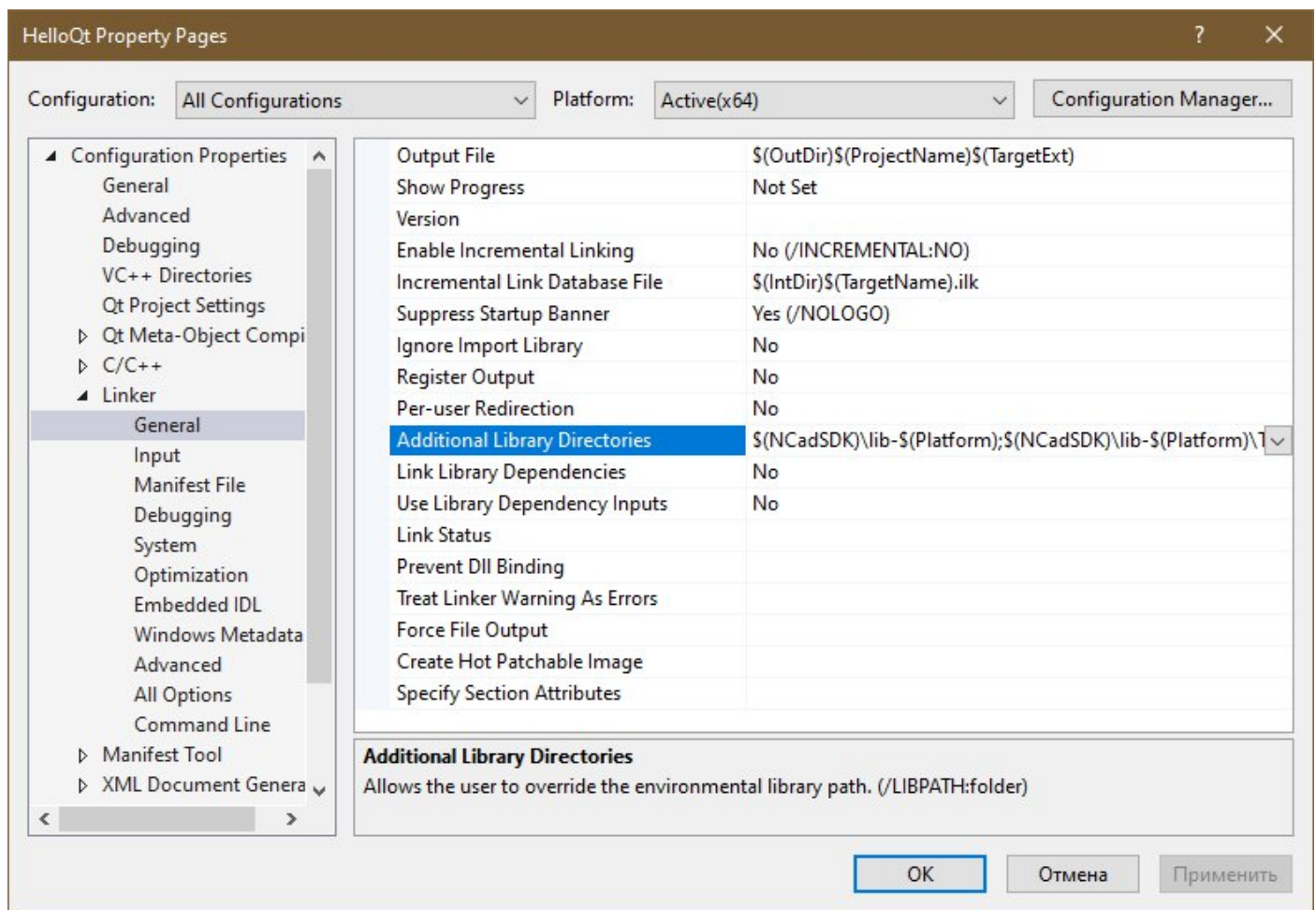


Рис. 25

\$(QtRoot)\lib

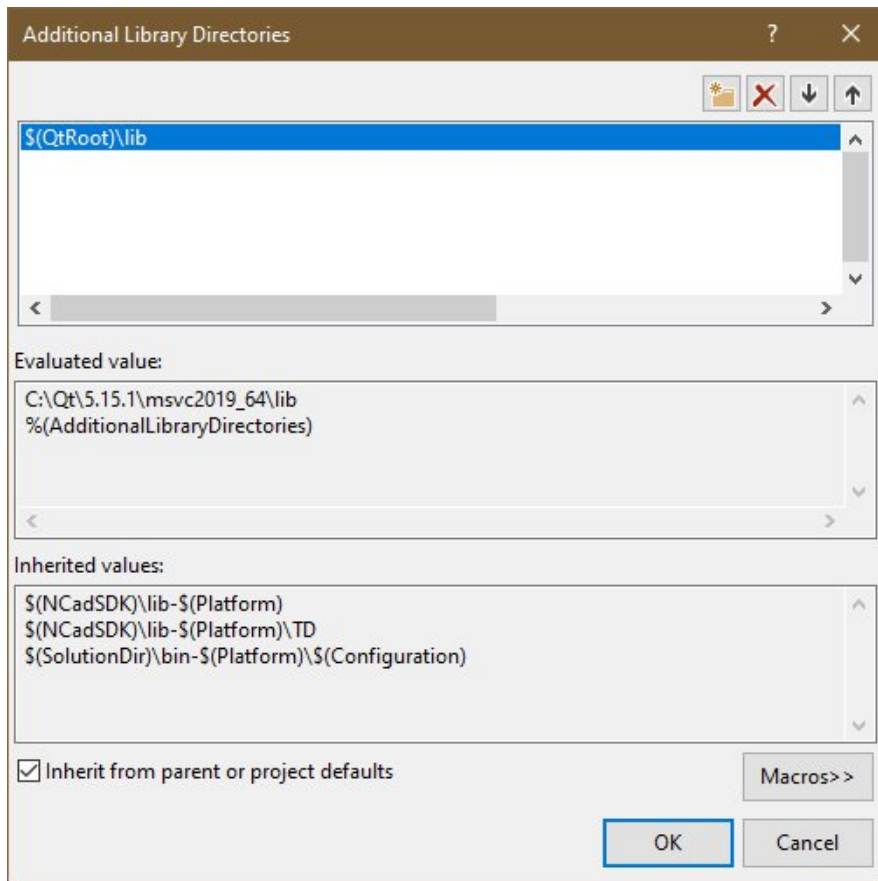


Рис. 26

## 8. Добавьте библиотеки

**Menu -> Project -> Properties -> Configuration Properties -> Linker -> Input -> Additional Dependencies**

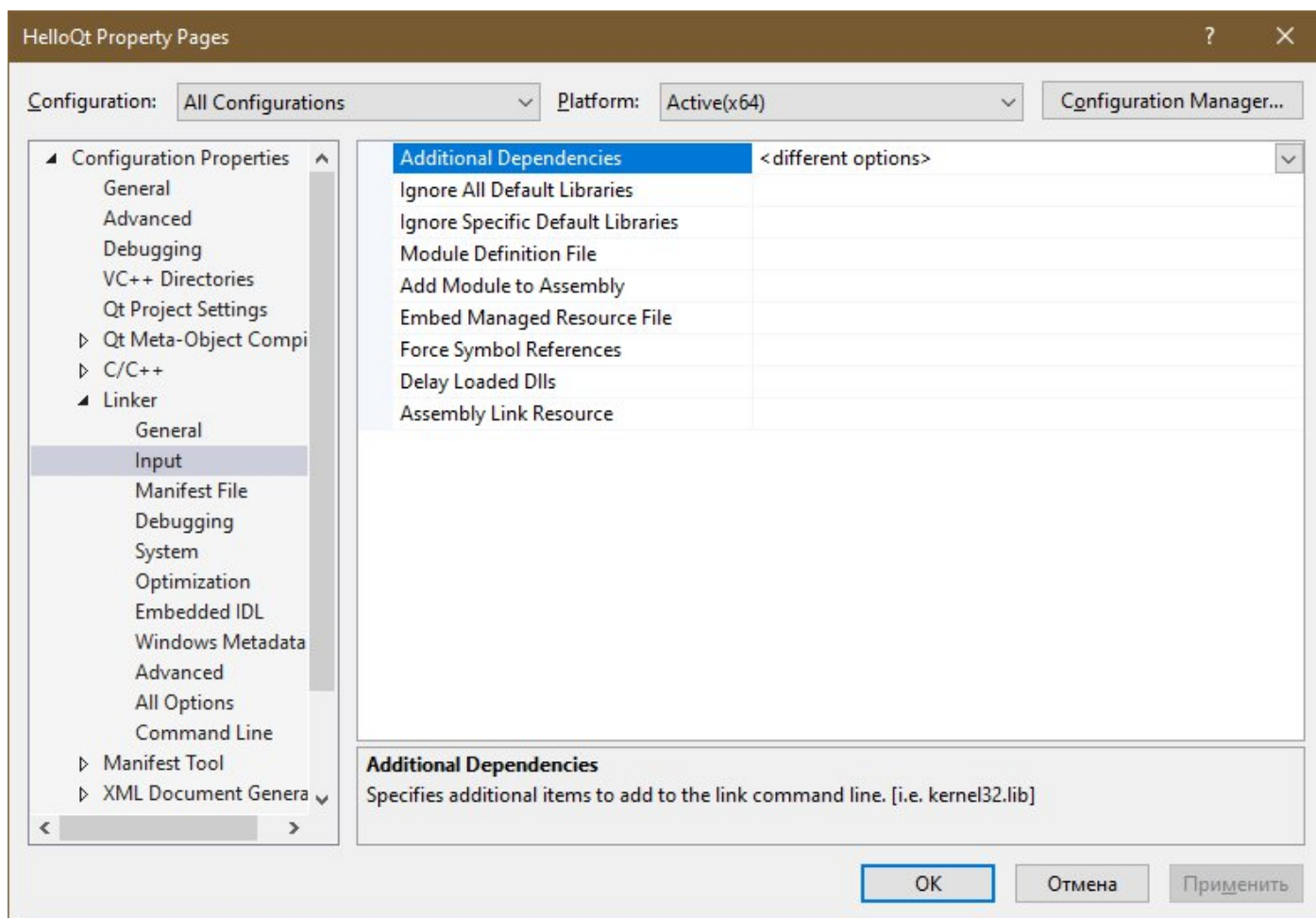


Рис. 27

Qt5Core.lib;Qt5Gui.lib;Qt5Widgets.lib;Qt5Xml.lib;Qt5WinExtras.lib;hostQt.lib;



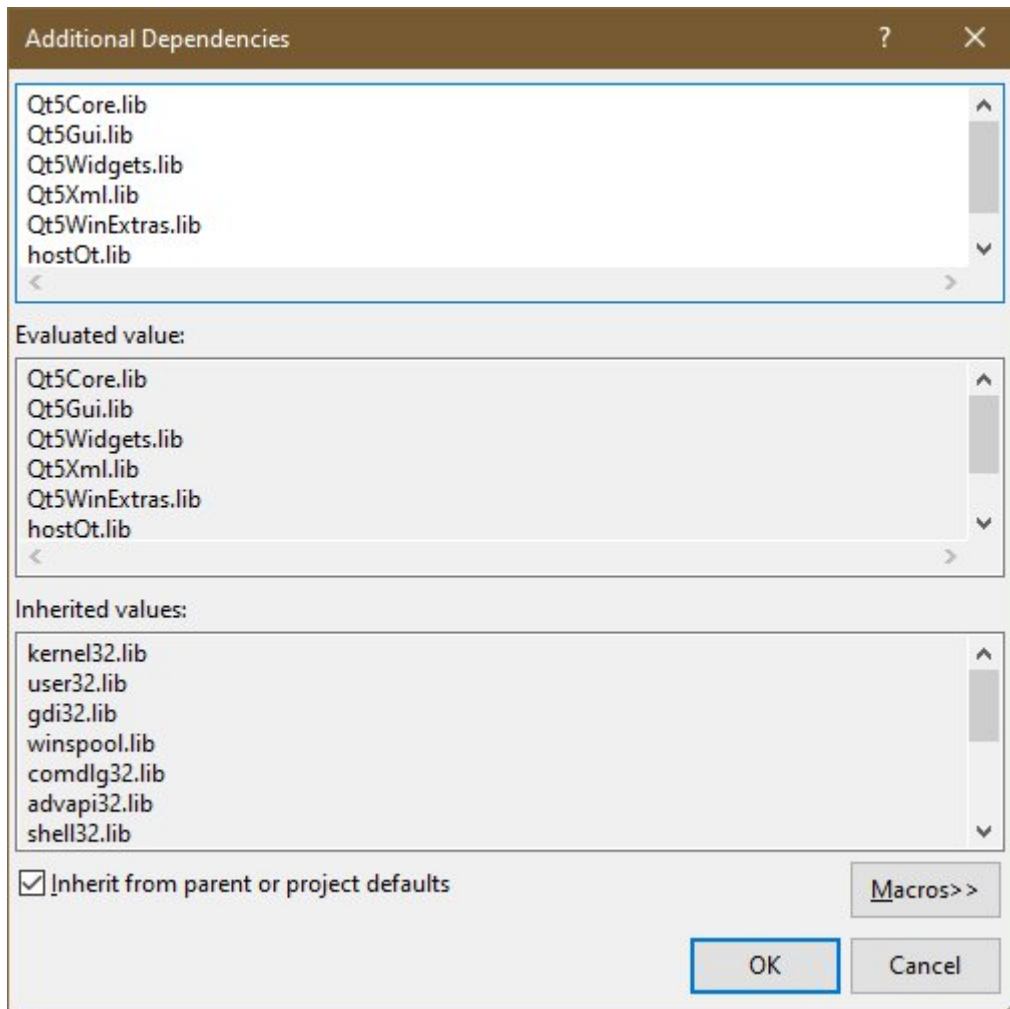


Рис. 28

9. Проверьте настройки и измените их, если требуется (для каждой платформы win32/x64 отдельно):

**Menu -> Project -> Properties -> Configuration Properties -> Advanced**

- Для Target File Extension установите значение .nrx;
- Для Use of MFC установите значение Use MFC in a Shared DLL;
- Для Character Set установите значение Use Unicode Character Set.

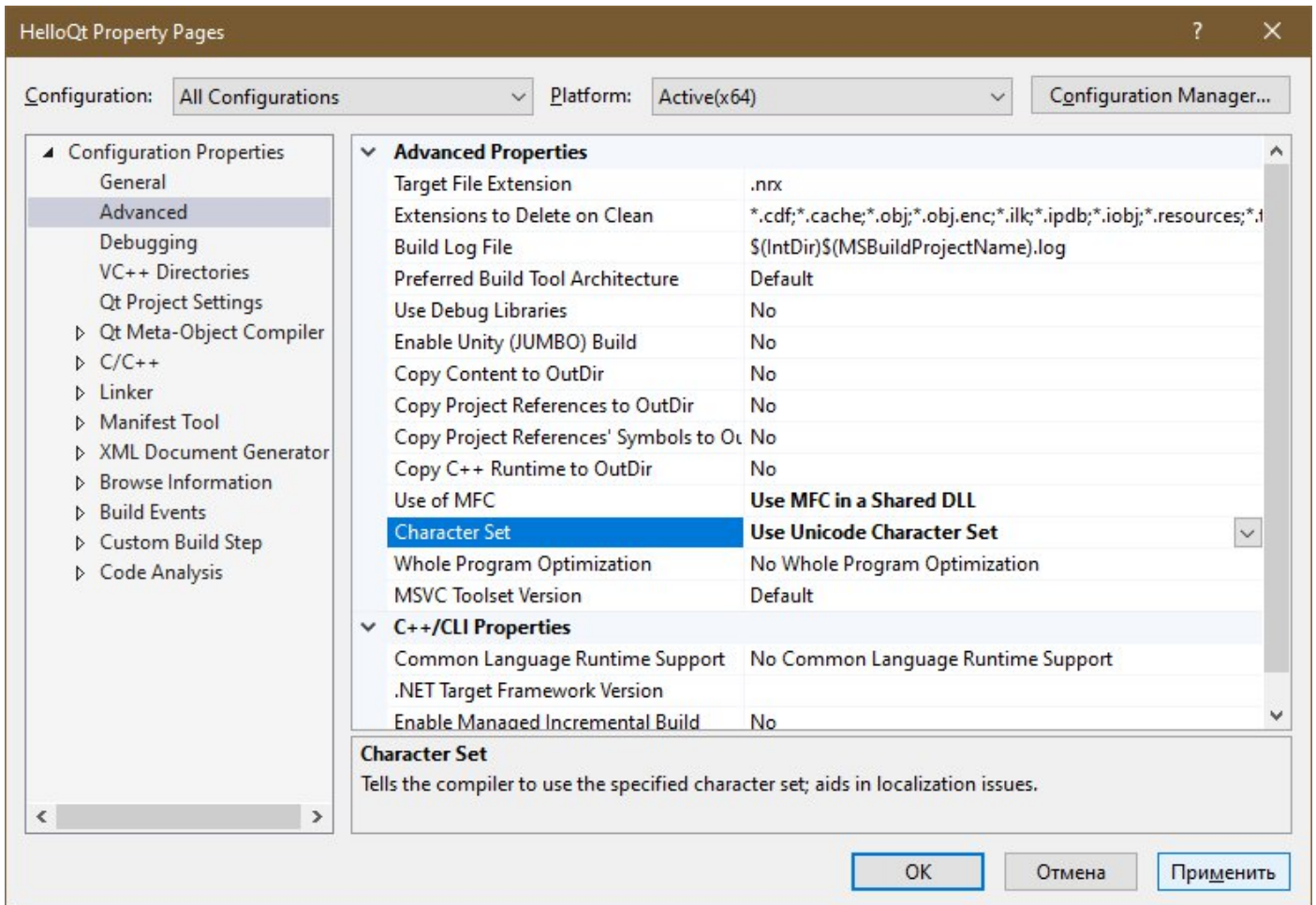


Рис. 29

10. Добавьте дополнительно флаг `/Zc:referenceBinding-`  
**Menu -> Project -> Properties -> Configuration Properties -> C/C++ -> Command Line -> Additional Options**

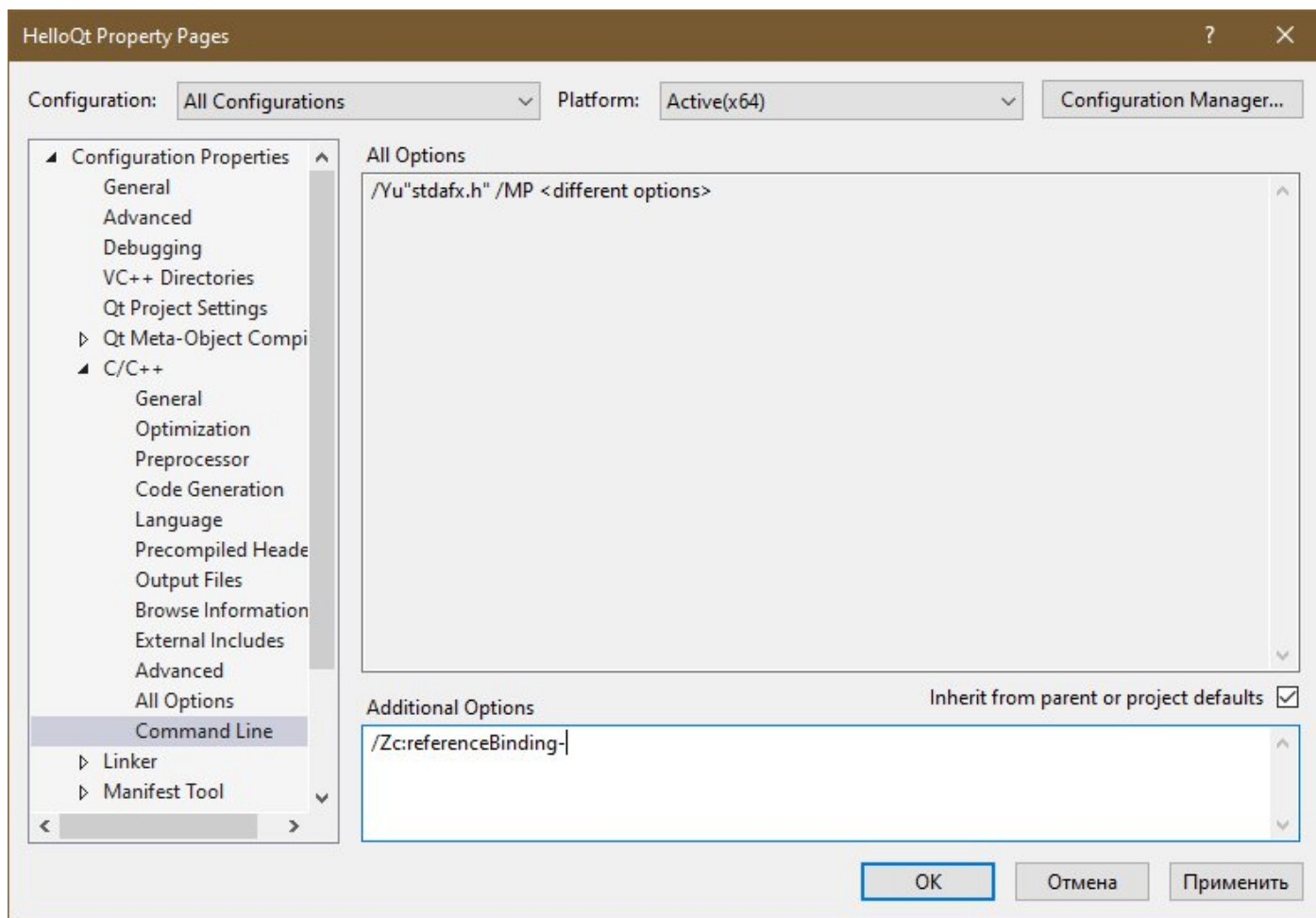


Рис. 30

10. Скопируйте файлы исходного кода из примера HelloQt, который находится в  $\$(NanoCadSDK)\ samples\NRX\HelloQt$ , где  $\$(NanoCadSDK)$  – путь до папки SDK nanoCAD, в папку с вашим проектом, заменив файлы **HelloQt.cpp**, **stdafx.cpp**, **stdafx.h**

11. Соберите проект.

12. Загрузите скомпилированный модуль в nanoCAD.

Команды приложения станут доступны в командной строке nanoCAD после загрузки.

Пример получившегося кода:

HelloQt.h

```
#pragma once

#include "helloqt_global.h"

class HELLOQT_EXPORT HelloQt
{
public:
    HelloQt();
};
```

## HelloQt.cpp

```
//
// Копирайт (С) 2019, ООО «Нанософт разработка». Все права защищены.
//
// Данное программное обеспечение, все исключительные права на него, его
// документация и сопроводительные материалы принадлежат ООО «Нанософт разработка».
// Данное программное обеспечение может использоваться при разработке и входить
// в состав разработанных программных продуктов при соблюдении условий
// использования, оговоренных в «Лицензионном договоре присоединения
// на использование программы для ЭВМ «Платформа nanoCAD»».
//
// Данное программное обеспечение защищено в соответствии с законодательством
// Российской Федерации об интеллектуальной собственности и международными
// правовыми актами.
//
// Используя данное программное обеспечение, его документацию и
// сопроводительные материалы вы соглашаетесь с условиями использования,
// указанными выше.
//

#include "stdafx.h"

#include "hostUI.h"
#include "hostQt.h"

extern "C" __declspec(dllexport) bool showDialog(HWND parent)
{
    auto win = new QWinWidget(parent);
    win->showCentered();

    QMessageBox::about(win, "QtTests.dll", "Hello, hostQt.dll (based on
qtwinmigrate\\examples\\qtdll!)");

    delete win;

    return TRUE;
}

void helloQtModalDlgCmd()
{
    acutPrintf(L"\nHello, hostQt.dll!\n");

    showDialog(adsw_acadMainWnd());
}

hostUiPaletteSet* m_pPalSet = NULL;

HINSTANCE _hdlInstance =NULL ;
AC_IMPLEMENT_EXTENSION_MODULE(theArxDLL);

class helloQtPalette : public hostQtPalette
{
    DECLARE_DYNAMIC(helloQtPalette)

public:
    helloQtPalette() {};

    afx_msg void OnSize(UINT nType, int cx, int cy)
    {
        if (m_pWinWidget)
        {
            HWND wnd = (HWND)m_pWinWidget->windowHandle()->winId();
            ::SetWindowPos(wnd, nullptr, 0, 0, cx, cy, SWP_NOZORDER);
        }
    }

    DECLARE_MESSAGE_MAP();
};
```

```

};

BEGIN_MESSAGE_MAP(helloQtPalette, hostQtPalette)
    //{AFX_MSG_MAP(helloQtPalette)
    ON_WM_SIZE()
    //}}AFX_MSG_MAP
END_MESSAGE_MAP()

IMPLEMENT_DYNAMIC(helloQtPalette, hostQtPalette);

void helloQtPaletteCmd()
{
    if (!m_pPalSet) {
        CAcModuleResourceOverride ThisRes;
        m_pPalSet = new hostUiPaletteSet();
        m_pPalSet->Create(L"Test Qt Palette Set", WS_CHILD | WS_DLGFRAME | WS_VISIBLE, CRect(30, 50,
270, 300),
            CWnd::FromHandle(adsw_acadMainWnd()), PSS_SNAP | PSS_PROPERTIES_MENU | PSS_AUTO_ROLLUP |
PSS_CLOSE_BUTTON);
        m_pPalSet->EnableDocking(CBRS_ALIGN_ANY);
        m_pPalSet->RestoreControlBar();

        helloQtPalette* pPal = new helloQtPalette();
        pPal->Create(WS_CHILD | WS_VISIBLE, L"Test Qt Palette1", m_pPalSet, 0);
        m_pPalSet->AddPalette(pPal);

        QWidget* pPaletteWidget1 = pPal->paletteWidget();

        QVBoxLayout* vbox = new QVBoxLayout(pPaletteWidget1);
        vbox->setSpacing(5);
        vbox->setMargin(6);
        QPushButton* pb = new QPushButton("Qt command button", pPaletteWidget1);
        pb->setObjectName("pb");
        vbox->addWidget(pb);
        QLabel* label = new QLabel("Some label", pPaletteWidget1);
        label->setObjectName("label");
        vbox->addWidget(label);
        QLineEdit* le1 = new QLineEdit();
        le1->setObjectName("le1");
        vbox->addWidget(le1);
        QLineEdit* le2 = new QLineEdit();
        le2->setObjectName("le2");
        vbox->addWidget(le2);
        QLineEdit* le3 = new QLineEdit();
        le3->setObjectName("le3");
        vbox->addWidget(le3);
        vbox->addStretch();
        //WinId winId = le3->winId(); // Make Qt windows real HWND windows

        pPaletteWidget1->setLayout(vbox);
        pPaletteWidget1->show();

        CRect cr;
        m_pPalSet->GetClientRect(&cr);
        pPal->OnSize(0, cr.Width(), cr.Height()); // Force to resize palette widget, needed when system
scale !=100%
    }
    else {
        m_pPalSet->Show(!m_pPalSet->IsWindowVisible());
    }
}

extern "C" __declspec(dllexport) AcRx::AppRetCode
acrxEEntryPoint(AcRx::AppMsgCode msg, void* appId)
{
    switch (msg)
    {
        case AcRx::kInitAppMsg:
            acrxDynamicLinker->unlockApplication(appId);
    }
}

```

```

    acrxDynamicLinker->registerAppMDIAware(appId);

    acedRegCmds->addCommand(L"HELLOQT_GROUP",
        L"_HELLOQTMODALDLG",
        L"HELLOQTMODALDLG",
        ACRX_CMD_TRANSPARENT,
        helloQtModalDlgCmd);

    acedRegCmds->addCommand(L"HELLOQT_GROUP",
        L"_HELLOQTPALETTE",
        L"HELLOQTPALETTE",
        ACRX_CMD_TRANSPARENT,
        helloQtPaletteCmd);

    break;

case AcRx::kUnloadAppMsg:
    acedRegCmds->removeGroup(L"HELLONRX_GROUP");

    if (m_pPalSet){
        m_pPalSet->DestroyWindow();
        m_pPalSet = 0;
    }

    break;
}

return AcRx::kRetOK;
}

```

## helloqt\_global.h

```

#pragma once

#include <QtCore/qglobal.h>

#ifdef BUILD_STATIC
# if defined(HELLOQT_LIB)
# define HELLOQT_EXPORT Q_DECL_EXPORT
# else
# define HELLOQT_EXPORT Q_DECL_IMPORT
# endif
#else
# define HELLOQT_EXPORT
#endif

```

## stdafx.h

```

//
// Copyright (C) 2019, 000 «Нанософт разработка». Все права защищены.
//
// Данное программное обеспечение, все исключительные права на него, его
// документация и сопроводительные материалы принадлежат 000 «Нанософт разработка».
// Данное программное обеспечение может использоваться при разработке и входит
// в состав разработанных программных продуктов при соблюдении условий
// использования, оговоренных в «Лицензионном договоре присоединения
// на использование программы для ЭВМ «Платформа nanoCAD»».
//
// Данное программное обеспечение защищено в соответствии с законодательством
// Российской Федерации об интеллектуальной собственности и международными
// правовыми актами.
//
// Используя данное программное обеспечение, его документацию и
// сопроводительные материалы вы соглашаетесь с условиями использования,
// указанными выше.
//

```

```

#pragma once

//-----
//- 'DEBUG workaround' below prevents the MFC or ATL #include-s from pulling
//- in "afx.h" that would force the debug CRT through #pragma-s.
#if defined(_DEBUG) && !defined(NC_FULL_DEBUG)
#define _DEBUG_WAS_DEFINED
#undef _DEBUG
#pragma message ("    Compiling MFC / STL / ATL header files in release mode.")
#endif

#pragma pack (push, 8)
#pragma warning(disable: 4786 4996)
//#pragma warning(disable: 4098)

//-----
#define STRICT

#ifndef VC_EXTRALEAN
#define VC_EXTRALEAN        //- Exclude rarely-used stuff from Windows headers
#endif

//- Modify the following defines if you have to target a platform prior to the ones specified below.
//- Refer to MSDN for the latest info on corresponding values for different platforms.
#ifndef WINVER
#define WINVER              //- Allow use of features specific to Windows 95 and Windows NT 4 or
later.
#define WINVER 0x0501      //- Change this to the appropriate value to target Windows 98 and
Windows 2000 or later.
#endif

#ifndef _WIN32_WINNT
#define _WIN32_WINNT 0x0501  //- Allow use of features specific to Windows NT 4 or later.
#define _WIN32_WINNT 0x0501  //- Change this to the appropriate value to target Windows 2000 or
later.
#endif

#ifndef _WIN32_WINDOWS
#define _WIN32_WINDOWS 0x0501  //- Allow use of features specific to Windows 98 or later.
#define _WIN32_WINDOWS 0x0501  //- Change this to the appropriate value to target Windows Me or
later.
#endif

#ifndef _WIN32_IE
#define _WIN32_IE 0x0501      //- Allow use of features specific to IE 4.0 or later.
#define _WIN32_IE 0x0501      //- Change this to the appropriate value to target IE 5.0 or later.
#endif

//-----
#define _ATL_APARTMENT_THREADED
#define _ATL_NO_AUTOMATIC_NAMESPACE
#define _ATL_CSTRING_EXPLICIT_CONSTRUCTORS    //- Some CString constructors will be explicit
//- Turns off ATL's hiding of some common and often safely ignored warning messages
#define _ATL_ALL_WARNINGS

//-----
#include <afxwin.h>
#include <afxext.h>
#include <AtlBase.h>
#include <AtlCom.h>
using namespace ATL ;

#include "dbxHeaders.h"
#include "AcExtensionModule.h"

#include <QMessageBox>
#include <QHBoxLayout>
#include <QLabel>
#include <QLineEdit>
#include <QPushButton>
#include <QWindow>

```

```
#pragma pack (pop)
```

```
//-----  
#ifdef _DEBUG_WAS_DEFINED  
#define _DEBUG  
#undef _DEBUG_WAS_DEFINED  
#endif
```

## stdafx.cpp

```
//  
// Копирайт (С) 2019, ООО «Нанософт разработка». Все права защищены.  
//  
// Данное программное обеспечение, все исключительные права на него, его  
// документация и сопроводительные материалы принадлежат ООО «Нанософт разработка».  
// Данное программное обеспечение может использоваться при разработке и входить  
// в состав разработанных программных продуктов при соблюдении условий  
// использования, оговоренных в «Лицензионном договоре присоединения  
// на использование программы для ЭВМ «Платформа nanoCAD»».  
//  
// Данное программное обеспечение защищено в соответствии с законодательством  
// Российской Федерации об интеллектуальной собственности и международными  
// правовыми актами.  
//  
// Используя данное программное обеспечение, его документацию и  
// сопроводительные материалы вы соглашаетесь с условиями использования,  
// указанными выше.  
//  
// stdafx.cpp : source file that includes just the standard includes  
// HelloNRX.pch will be the pre-compiled header  
// stdafx.obj will contain the pre-compiled type information  
  
#include "stdafx.h"  
  
// TODO: reference any additional headers you need in STDAFX.H  
// and not in this file
```