

# Как дополнить компиляцией метаданных проект Visual Studio с примерами из SDK nanoCAD с помощью расширения Qt VS Tools.

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

## 1. Установите Qt VS Tools

В Visual Studio (Menu -> Extensions -> Manage Extensions)

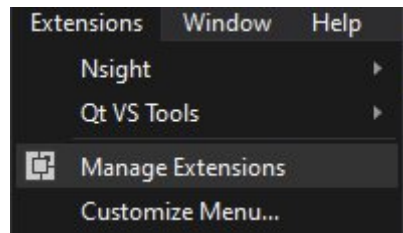


Рис. 1

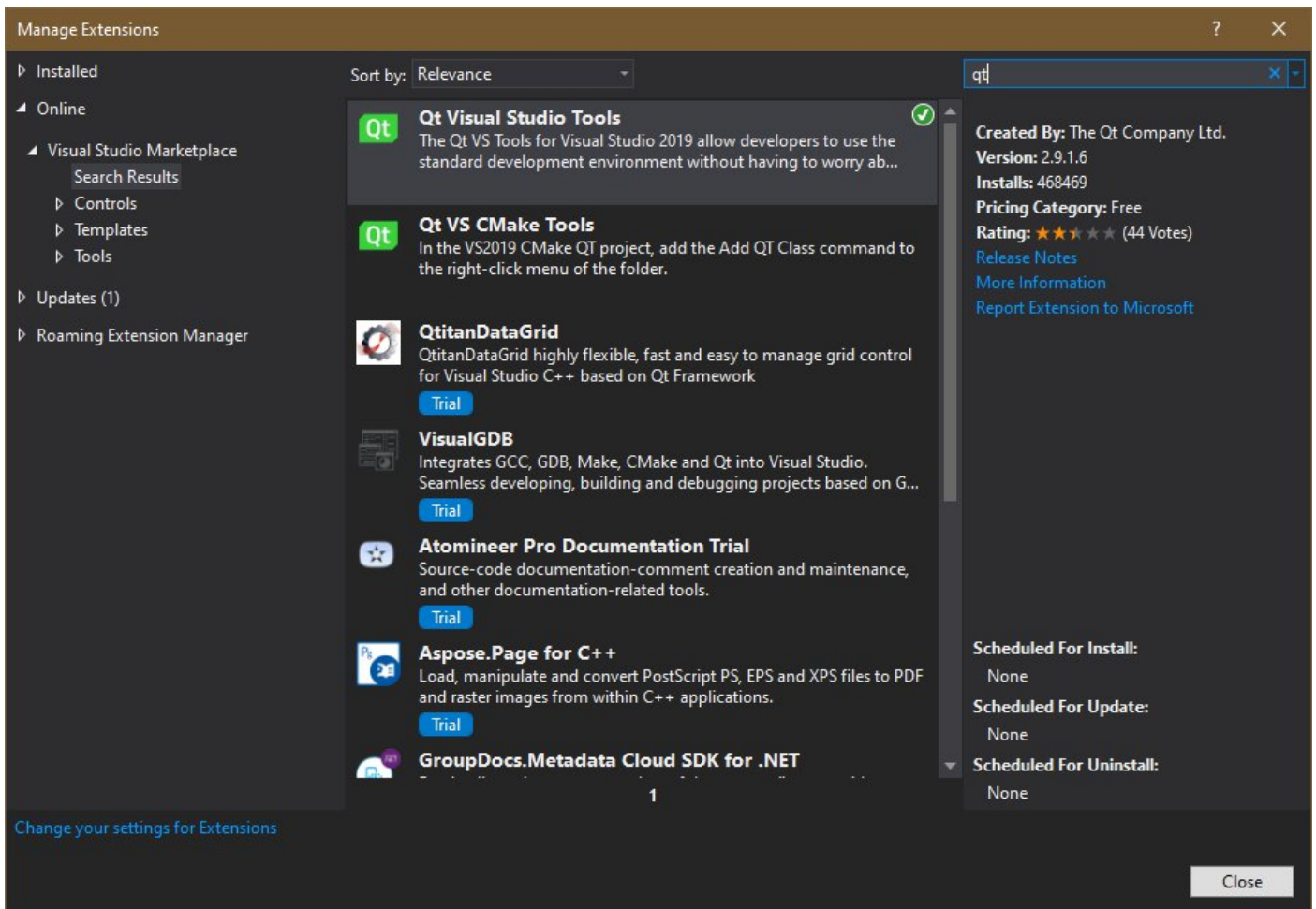


Рис. 2

## 2. Добавьте версию Qt в расширение Qt Vs Tools

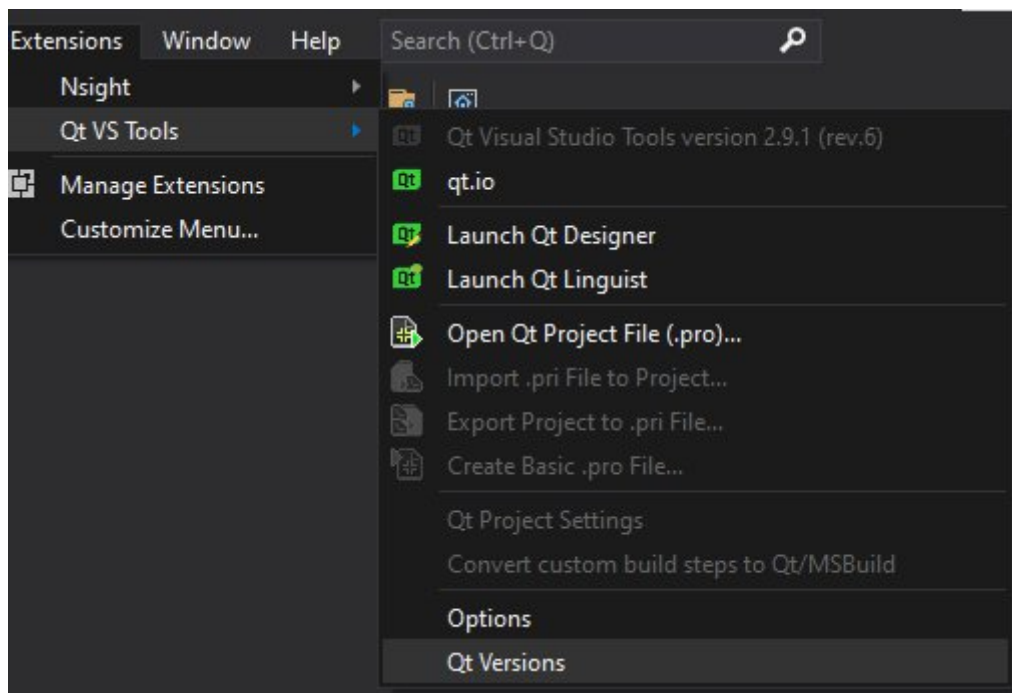


Рис. 3

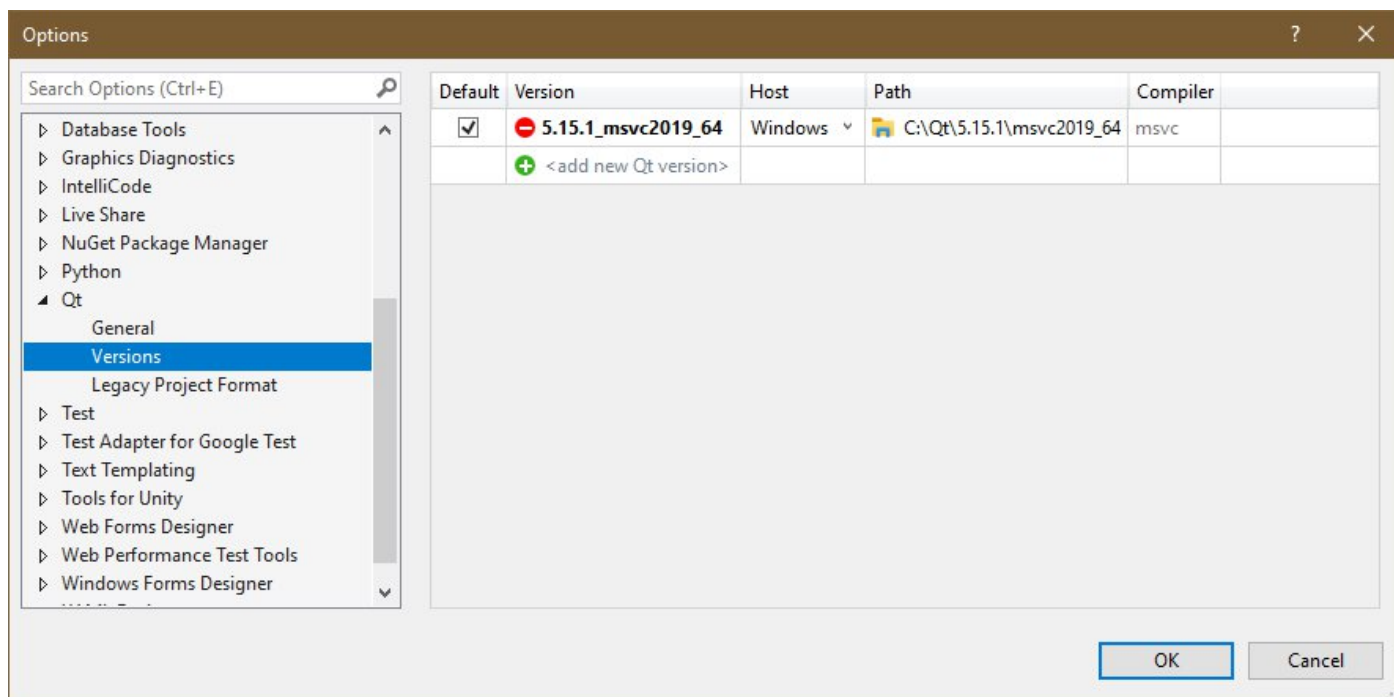


Рис. 4

3. Добавьте переменную среды QtToolsPath = C:\Qt\5.15.1\msvc2019\_64\bin и QtRoot = C:\Qt\5.15.1\msvc2019\_64

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

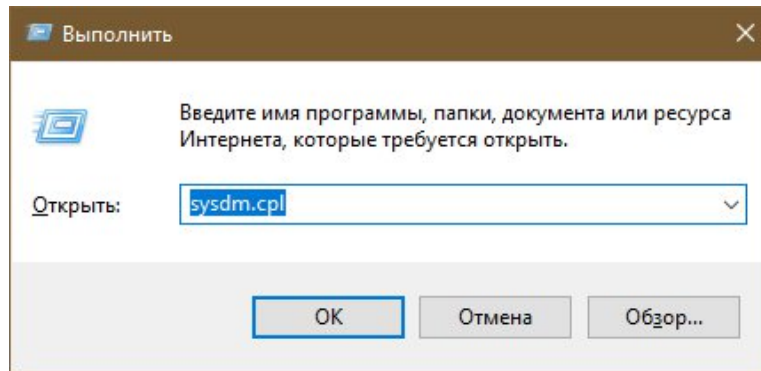


Рис. 5

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

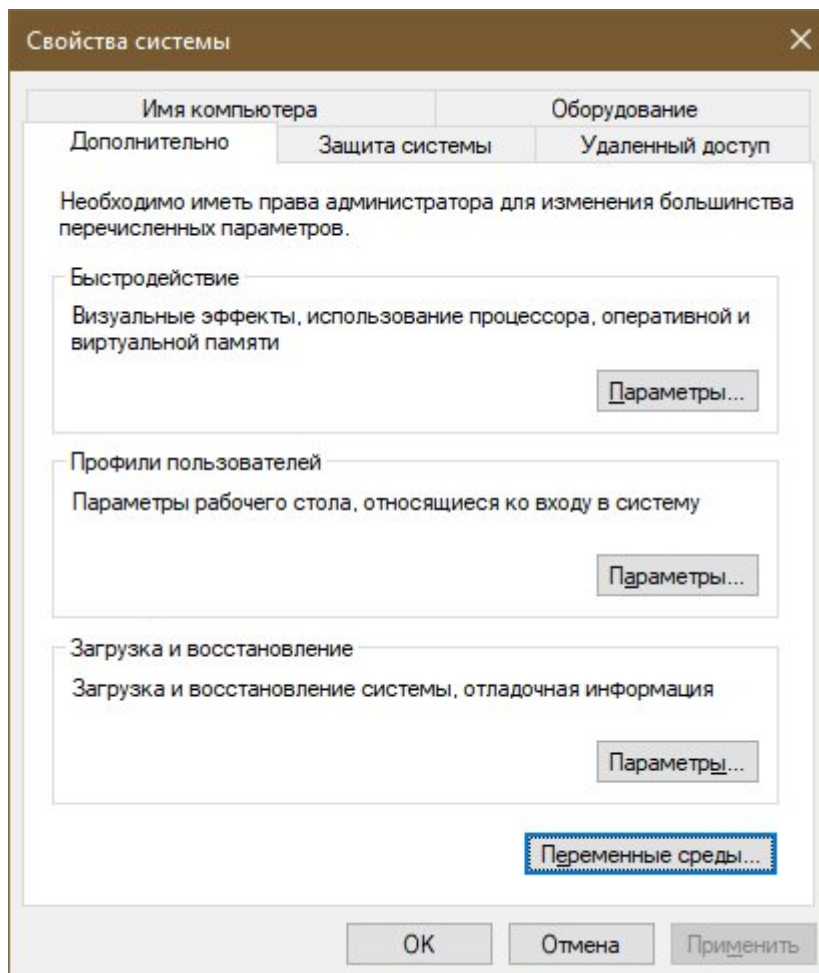


Рис. 6

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

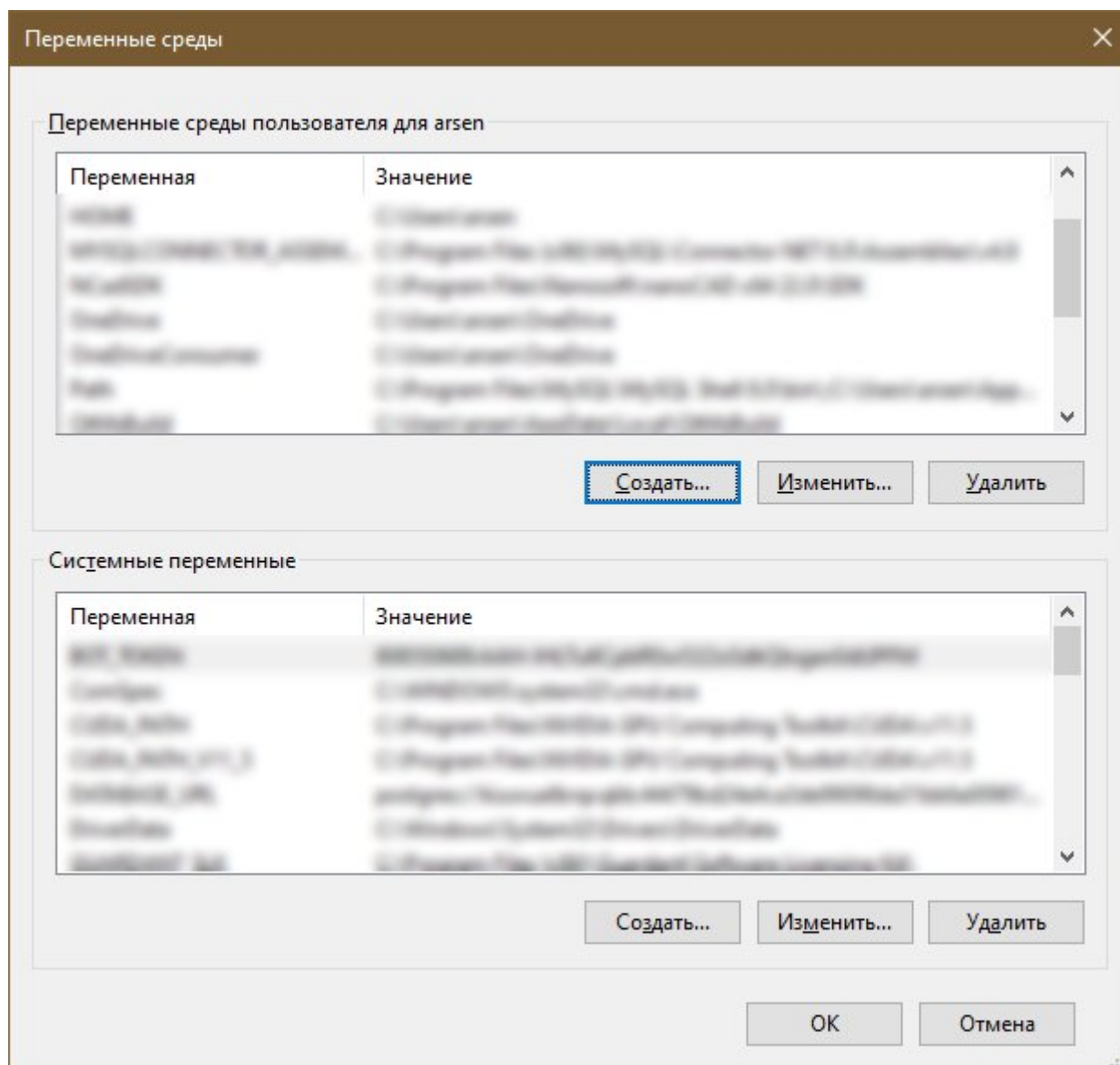


Рис. 7

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.

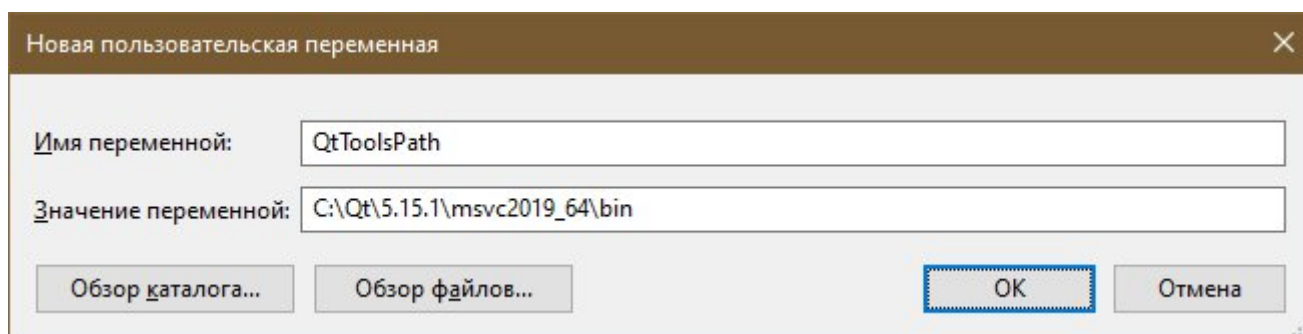


Рис. 8

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

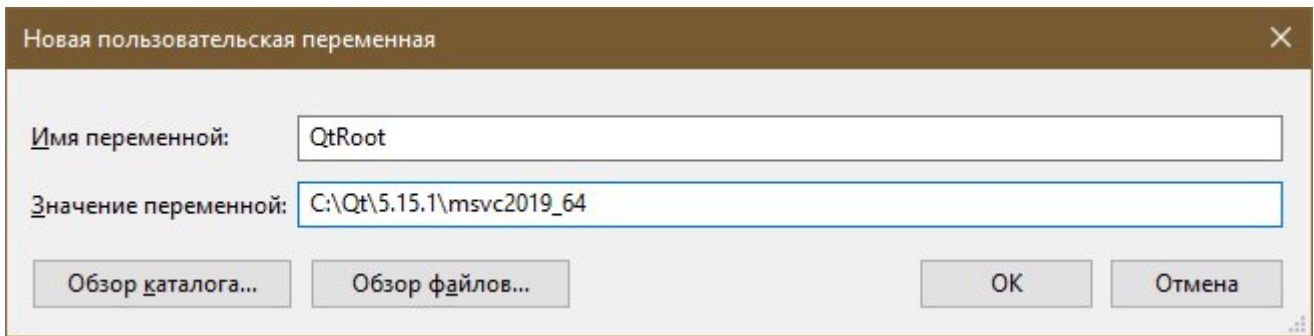


Рис. 9

4. В файле .vcxproj целевого проекта замените строчку `<keyword>...</keyword>` на `<keyword>QtVS_v304</keyword>` в разделе `<PropertyGroup Label="Globals">`

Можно либо отредактировать его в текстовом редакторе, либо открыть проект в Visual Studio, нажать по нему правой кнопки мыши и выбрать пункт Unload Project.

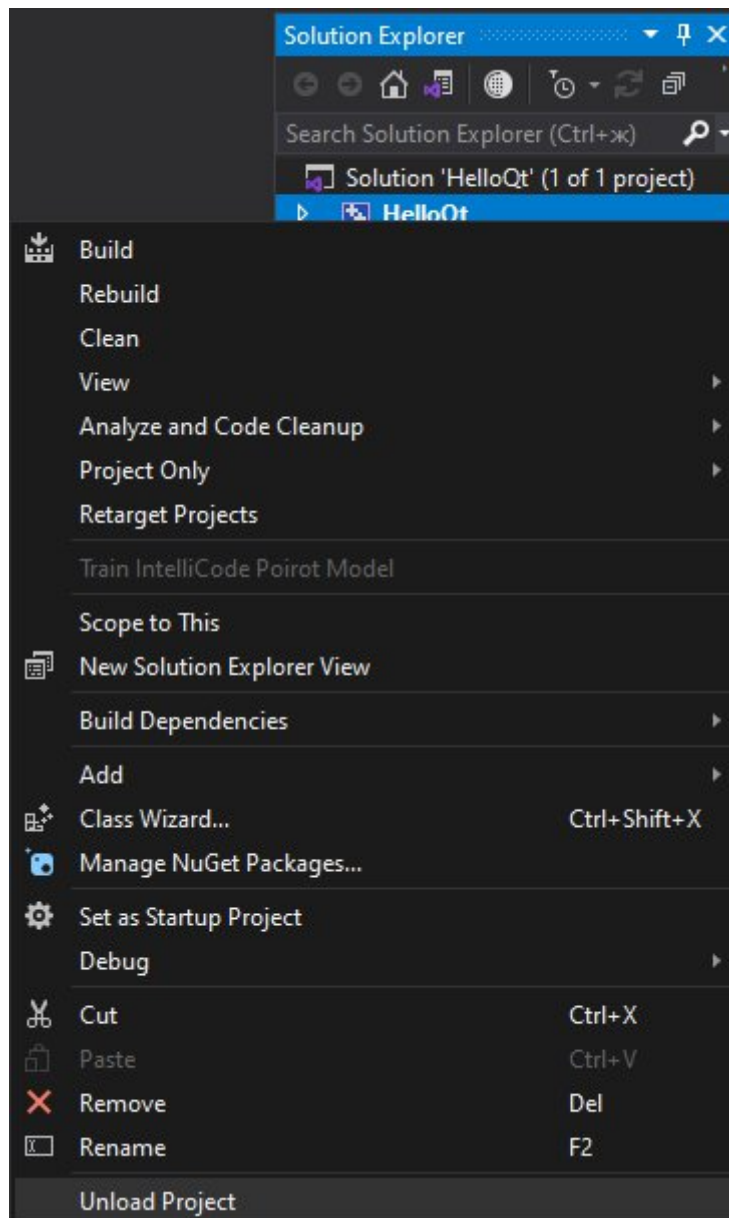


Рис. 10



Затем отредактировать файл в среде Visual studio

```
HelloQt.vcxproj  ▸ X
24  <PropertyGroup Label="Globals">
25  <ProjectGuid>{E9437F42-2BB7-495D-82A1-B480D98C34C9}</ProjectGuid>
26  <Keyword>QtVS_v304</Keyword>
27  <RootNamespace>HelloQt</RootNamespace>
28  <WindowsTargetPlatformVersion>10.0</WindowsTargetPlatformVersion>
29  </PropertyGroup>
```

Рис. 11

и нажать Reload Project в контекстном меню проекта

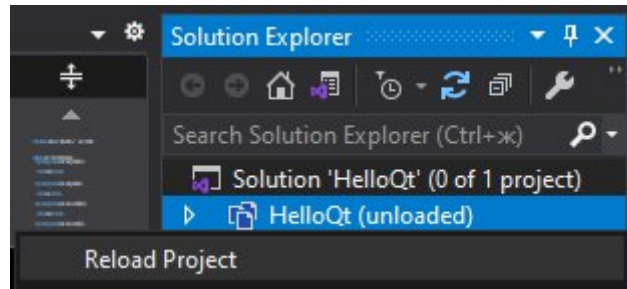


Рис. 12

5. Откройте проект в Visual Studio и в расширении Qt VS Tools выберите Convert custom build steps to Qt/MSBuild

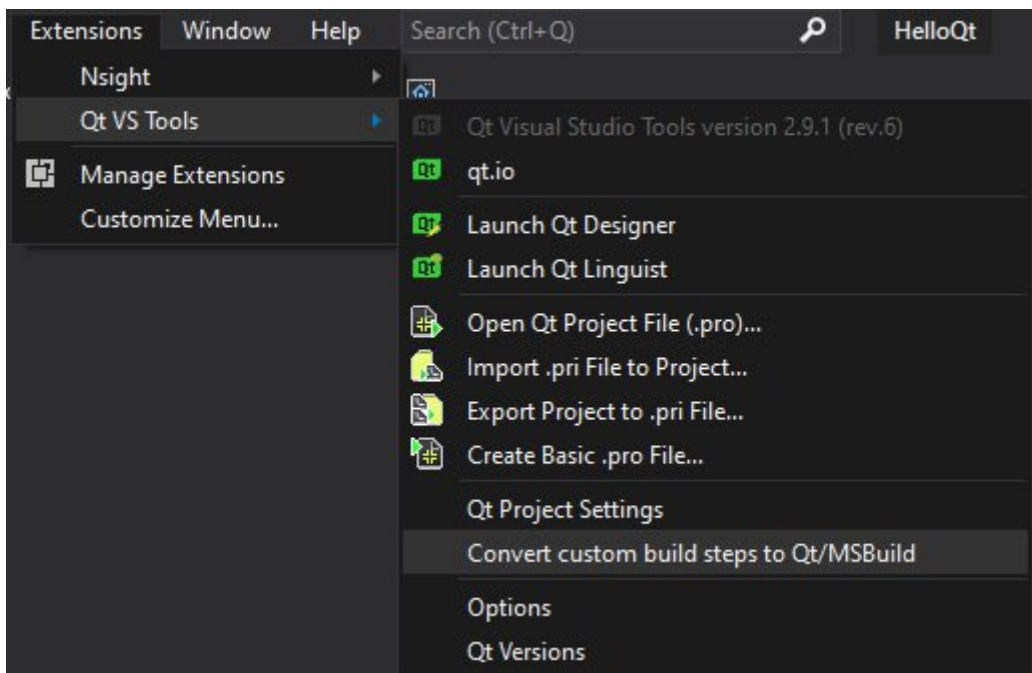


Рис. 13

В появившемся диалоговом окне нажмите «Да»

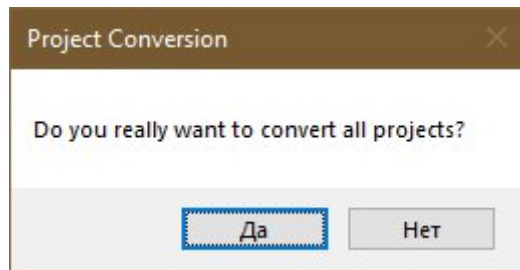


Рис.14

6. Добавьте элемент «класс Qt Widget с макросом Q\_ОБЪЕКТ» и сохранить файл  
**Menu -> Project -> Add ->NewItem...**

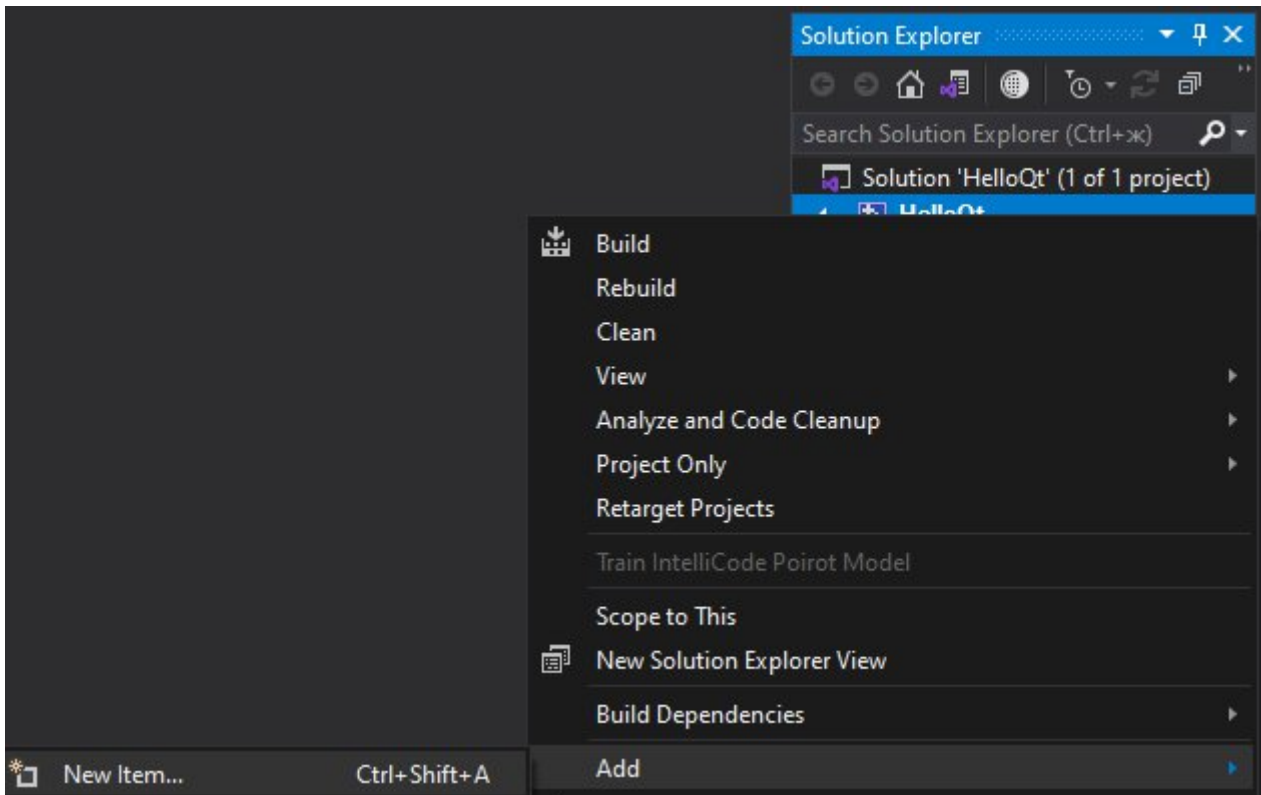


Рис. 15

**Qt -> Qt Widgets Class -> Add**

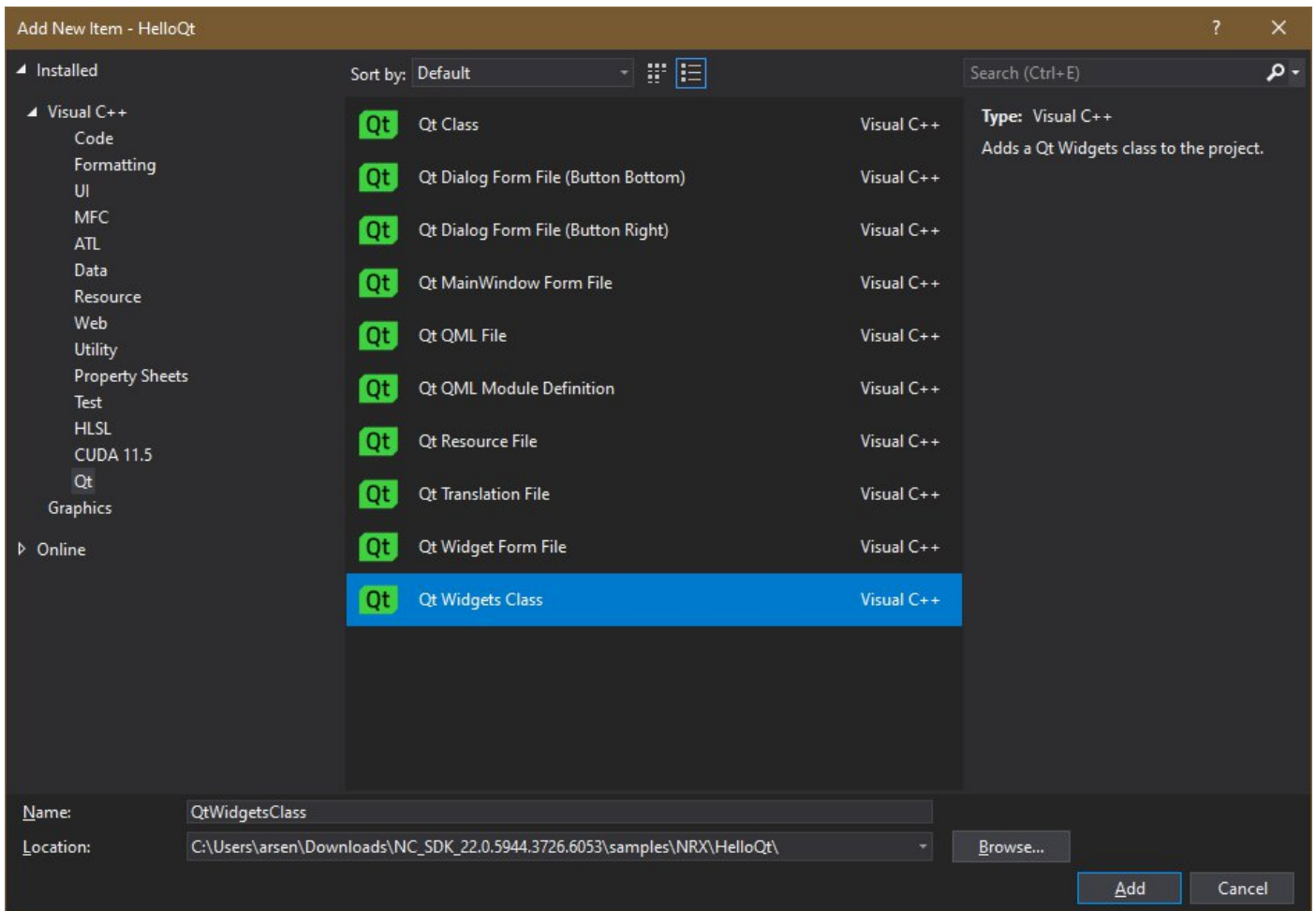


Рис. 16

Нажмите Next



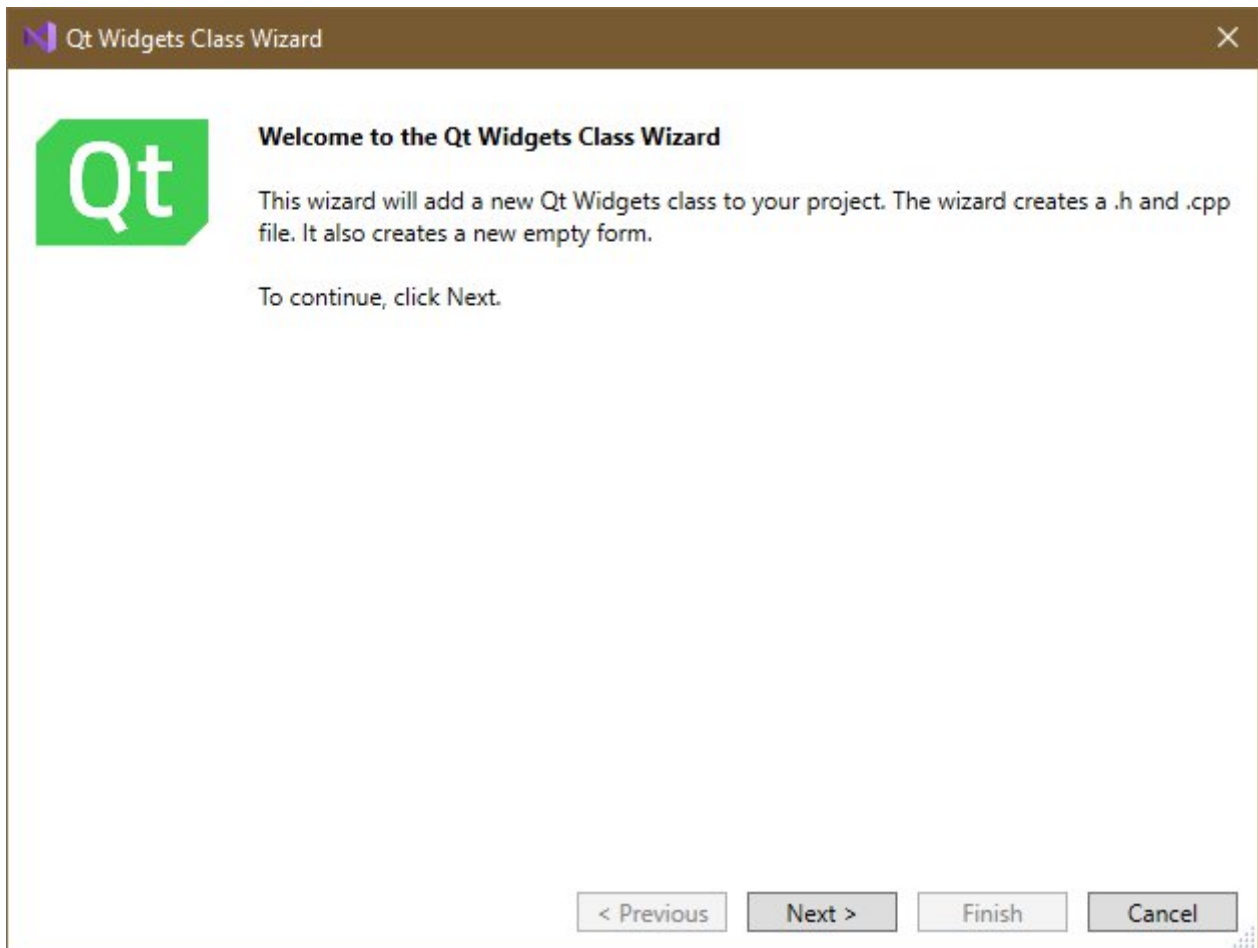


Рис. 17

Настройте класс и нажмите **Finish**

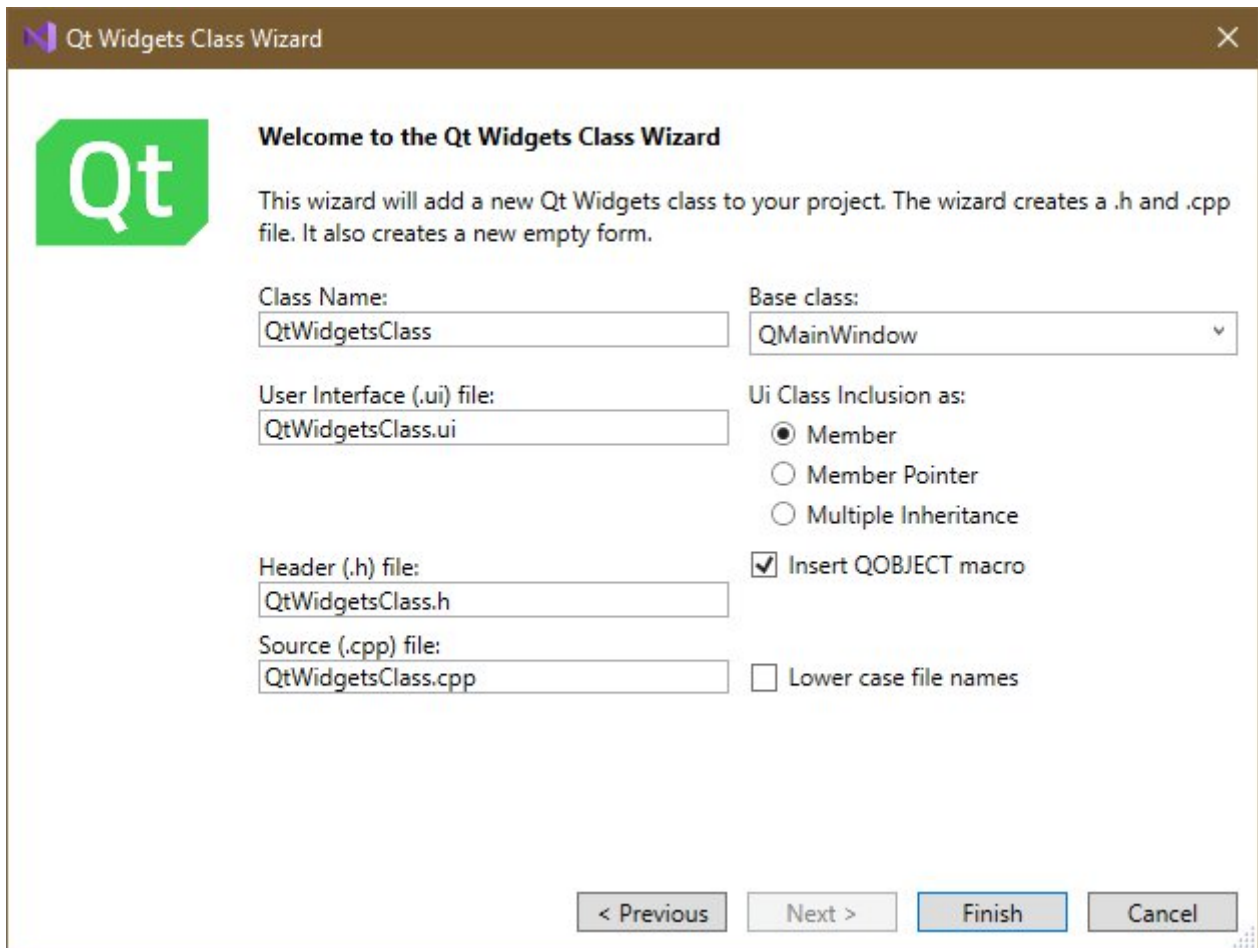


Рис. 18

7. Добавьте флаг /Zc:referenceBinding-

## Menu -> Project -> Properties -> Configuration Properties -> C/C++ -> General -> Additional Options

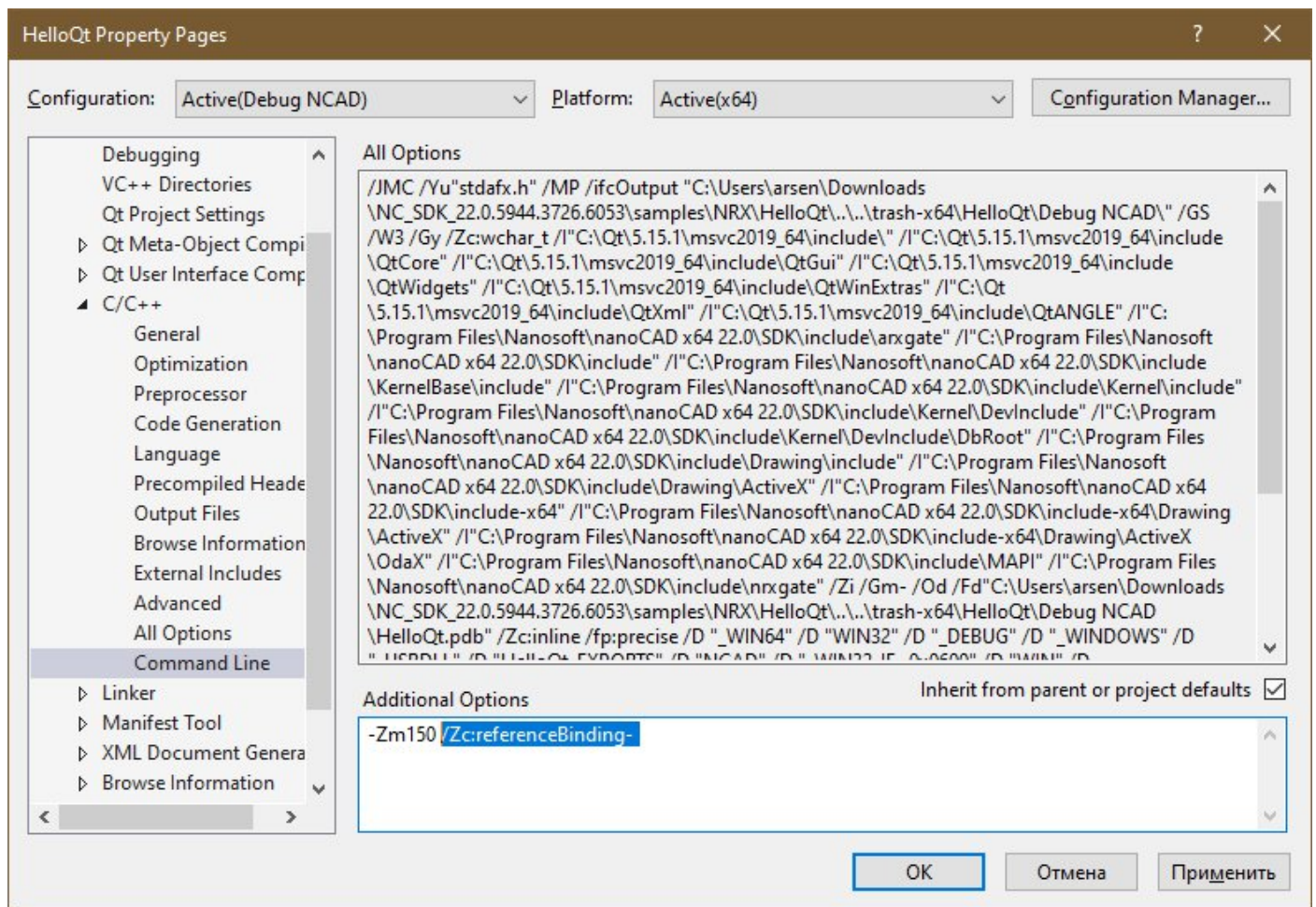


Рис. 19

8. Соберите проект.

9. Загрузите скомпилированный модуль в nanoCAD.

Команды приложения станут доступны в командной строке nanoCAD после загрузки.

Пример получившегося кода:

### QtWidgetsClass.h

```
#pragma once

#include <QMainWindow>
#include "ui_QtWidgetsClass.h"

class QtWidgetsClass : public QMainWindow
{
    Q_OBJECT

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

private:
    Ui::QtWidgetsClassClass ui;
};
```

### QtWidgetsClass.cpp

```

#include "stdafx.h"
#include "QtWidgetsClass.h"

QtWidgetsClass::QtWidgetsClass(QWidget *parent)
    : QMainWindow(parent)
{
    ui.setupUi(this);
}

QtWidgetsClass::~QtWidgetsClass()
{}

```

## stdafx.h

```

//
// Копирайт (С) 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 входить
// в состав разработанных программных продуктов при соблюдении условий
// использования, оговоренных в «Лицензионном договоре присоединения
// на использование программы для ЭВМ «Платформа папоCAD»».
//
// 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                //- 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          //- 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          //- 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              //- 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>

#include "QtWidgetsClass.h"

#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
```

## HelloQt.cpp

```
//
// Копирайт (С) 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 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 _hdllInstance =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"HELLOQT_GROUP");

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

        break;
    }

    return AcRx::kRetOK;
}

```