Публикация приложения на десктопных платформах

При запуске проекта Qt (например, в Qt Creator) фреймворк создает папку с бинарным файлом приложения. Однако создаваемый файл содержит много служебной информации, которая применяется для разработки и отладки, но простому пользователю не нужна. Кроме того, для полноценной работы приложению нужны дополнительные файлы - различные библиотеки, которые применяются для его работы. В процессе отладки и запуска приложения в Qt Creator такие файлы автоматически связываются с бинарным файлом. Однако в самой папке сгенерированного бинарного файла эти файлы могут отсутствовать, а бинарный файл при запуске может их не найти. В связи с этим возникает необходимость создания выходного пакета приложения, которое можно просто скопировать на другой компьютер с определенной операционной системой и одним нажатием на бинарный файл, запустить его. Рассмотрим, как собрать приложение на Qt в выходной пакет.

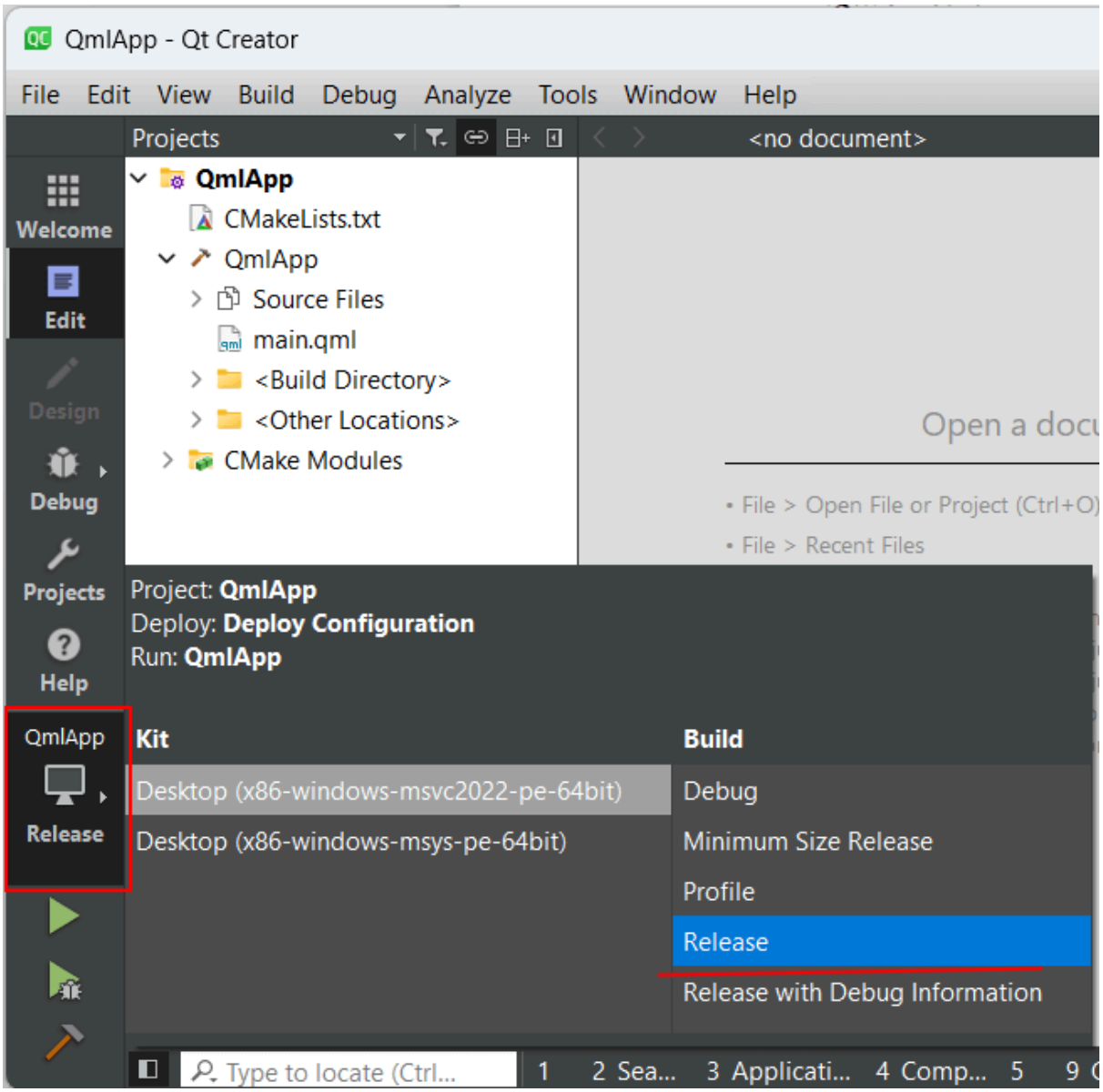
Настройка информации приложения

Перед сборкой мы можем добавить для приложения некоторые метаданные о приложении, в частности, имя, версию приложения, сведения о компании. Для этого в файле main.cpp у класса QApplication (или QGuiApplication в случае с qml) можно использовать несколько методов

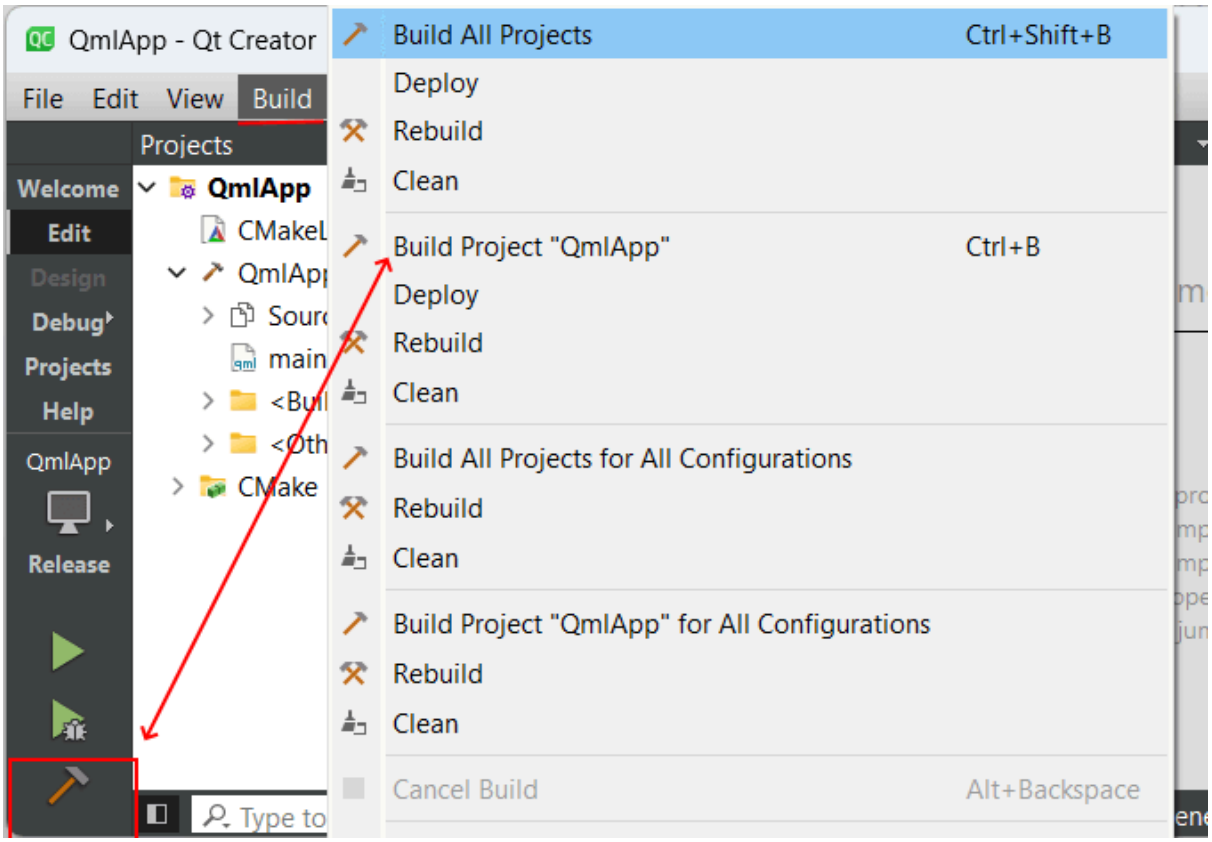
```
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    app.setOrganizationName("MyCorp");              // имя компании
    app.setOrganizationDomain("metanit.com");       // домен компании
    app.setApplicationName("Test Application");     // имя приложения
    app.setApplicationVersion("1.0.0");             // версия приложения

    //  остальное содержимое файла
```

Построение приложения в режиме Release

Для создания релизной версии приложения для проекта надо в качестве режима построения указать Release (Выпуск).

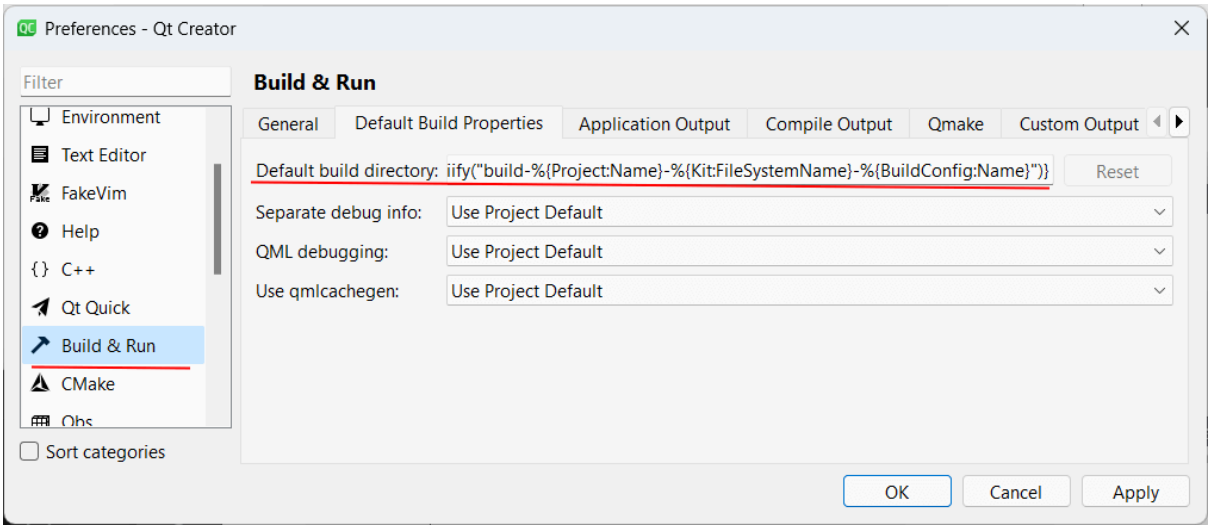


Затем по нажатию на пункт меню Build можно выбрать один из пунктов контекстного меню для построения проекта (например, на пункт Build Project/Rebuild)



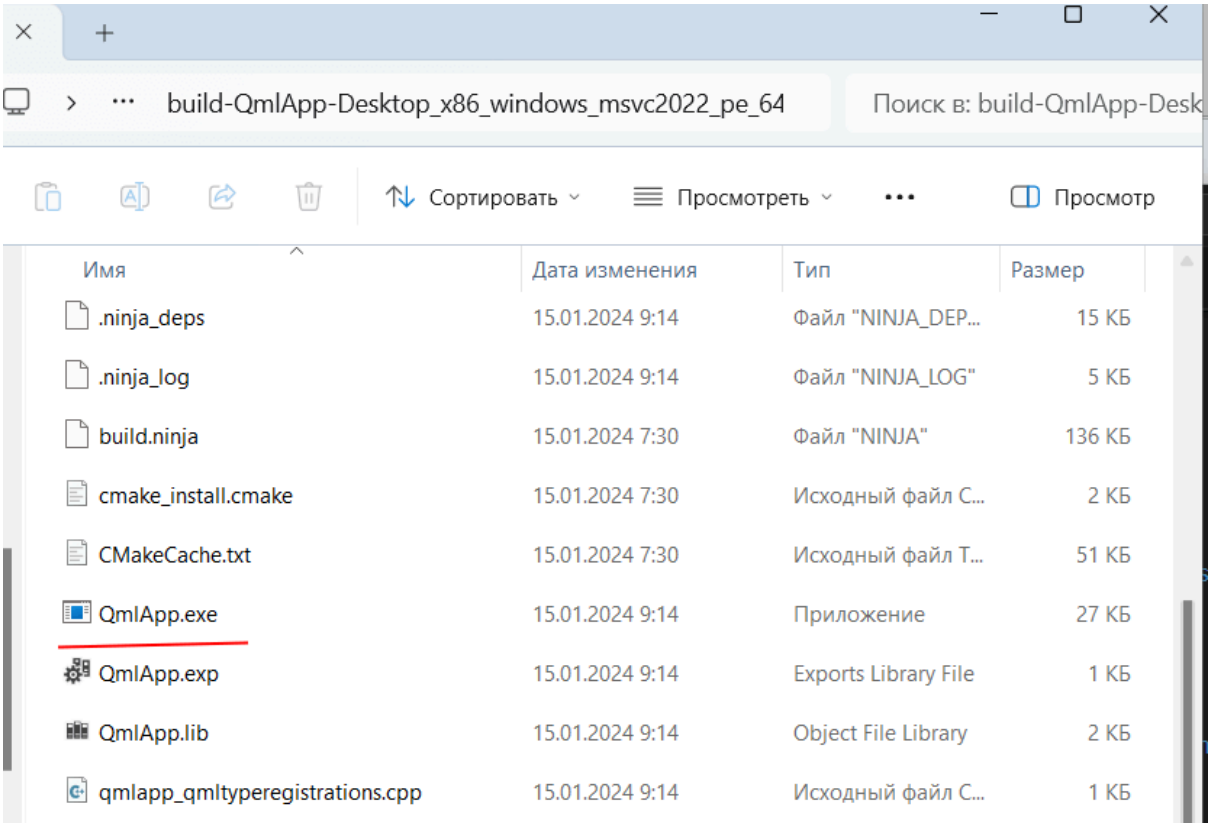
Также для построения можно нажать на значок молоточка в левом нижнем углу Qt Creator.

Через пункт меню Edit -> Preferences (Инструменты -> Параметры) можно перейти к окну настроек и далее выбрать пункт Build & Run для конфигурации ряд настроек построения. В частности, мы можем задать выходной каталог для построения приложения

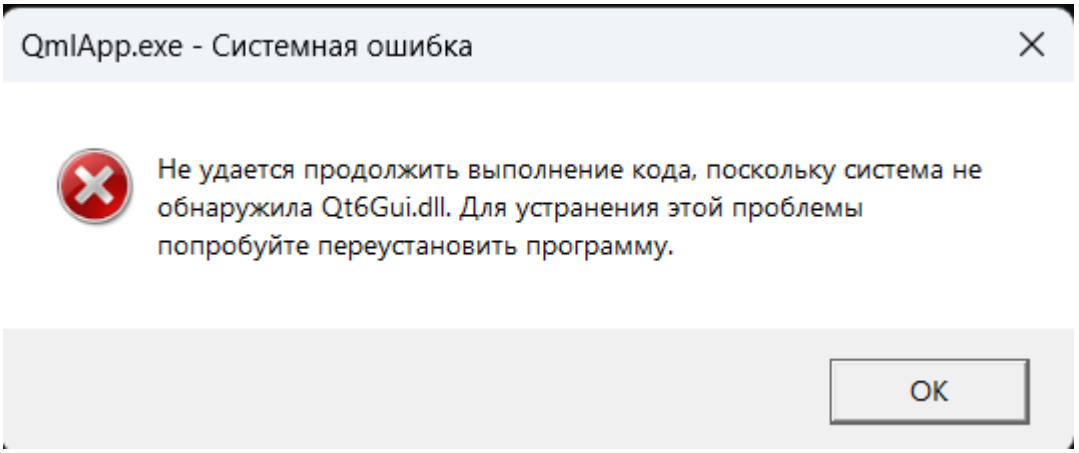


По умолчанию приложение создается в текущем каталоге проектов в папку, которая называется по шаблону build-{Имя_Проекта}-{инструментарий_построения}-{Имя_конфигурации}")

Например, в моем случае проект называется "QmlApp", инструментарий построения (Kit) - "Desktop_x86_windows_msvc2022_pe_64bit", а конфигурация - "Release", соответственно папка приложения называется build-QmlApp-Desktop_x86_windows_msvc2022_pe_64bit-Release. Здесь я могу найти исполняемый файл построенного приложения, который по умолчанию называется по имени проекта:



Однако при попытке запуска подобного файла мы можем столкнуться с ошибкой отстутствия некоторых библиотек:



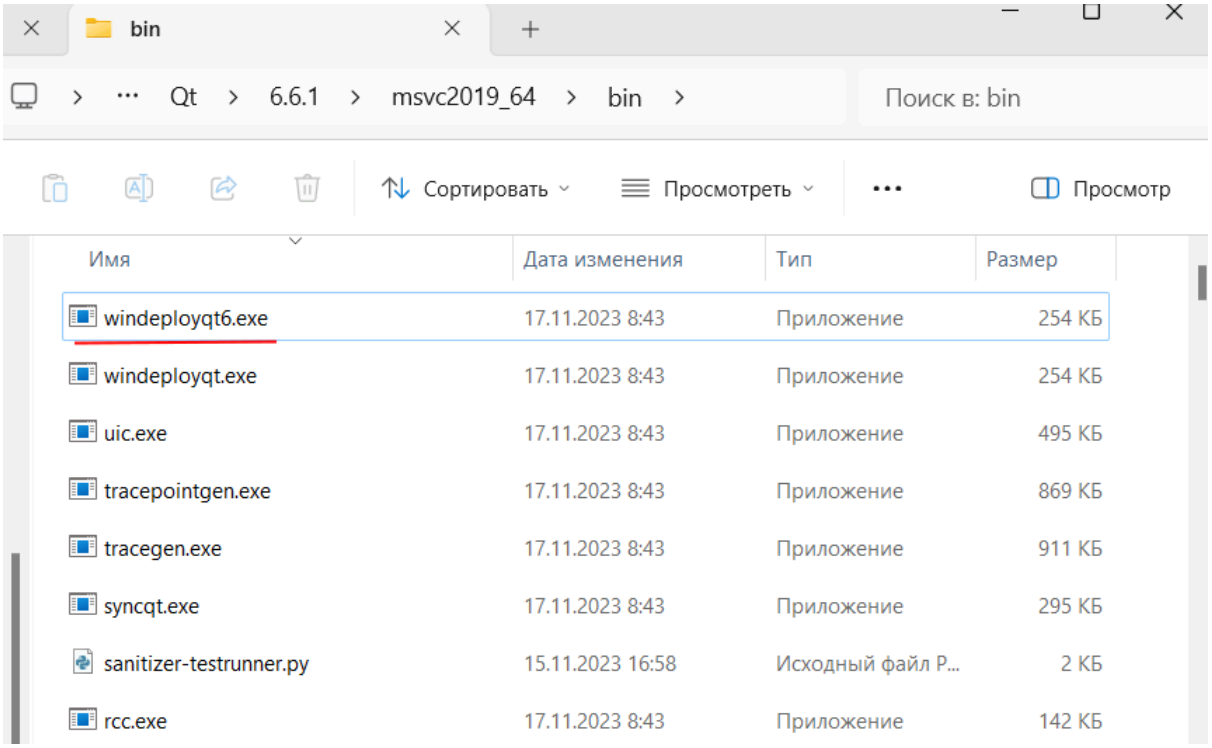
Мы могли бы вручную попытаться скопировать все необходимые файлы в каталог приложения. Однако фреймворк Qt предоставляет ряд утилит для автоматизации публикации приложения.

Публикация на Windows

Для публикации приложения на Windows Qt предоставляет специальную утилиту windeployqt6, которая находится в папке фреймворка (для старых версий фреймворка - Qt 5 также поставляется утилита windeployqt.exe - без цифры 6 в названии).

У меня:

D:\QT\5.12.12\mingw73_64\bin



В общем случае ее применение имеет следующий синтаксис:

windeployqt6 [options] [files]

Если публикуется приложение на виджетах Qt, то этой утилите достаточно передать путь к приложению:

windeployqt6 C:\Users\eugen\Documents\QtProjects\build-QmlApp-Desktop_x86_windows_msvc2022_pe_64bit-Release\QmlApp.exe

Но если публикуется приложение, которое использует QML/Qt Quick, то этой утилите передается аргумент --qmldir, путь к папке qml-файлов и путь к файлу приложения:

windeployqt6 --qmldir путь_к_файлам_qml путь_к_приложению

Например, в моем случае папка проекта, где размещены файлы qml - "C:\Users\eugen\Documents\QtProjects\QmlApp", а путь к построенному приложению - "C:\Users\eugen\Documents\QtProjects\build-QmlApp-Desktop_x86_windows_msvc2022_pe_64bit-Release\QmlApp.exe", поэтому в моем случае команда на публикаю выглядит так:

windeployqt6 --qmldir C:\Users\eugen\Documents\QtProjects\QmlApp C:\Users\eugen\Documents\QtProjects\build-QmlApp-Desktop_x86_windows_msvc2022_pe_64bit-Release\QmlApp.exe

В результате windeployqt6 скопирует все необходимые файлы в папку приложения. В дальнейшем мы сможем переносить папку приложения с компьютера на компьютер с той же ОС и запускать приложение.

Минусом такого подхода является то, что большой размер папки даже для примитивного приложеия и что при этом создается и компируется много каталогов и файлов, которые в реальности не нужны для работы приложения.

Публикация на Linux

Qt не предоставляет для дистрибутивов Linux готовых инструментов, аналогичных windeployqt. Возможно, это связано с большим количеством разновидностей Linux. Но есть аналогичный неофициальный инструмент с открытым исходным кодом под названием linuxdeployqt. Он также принимает скомпилированный файл приложения в качестве параметра и превращает его в автономный пакет путем репликации ресурсов проекта в пакет. Пользователи могут получить сгенерированный пакет. linuxdeployqt

может упаковывать определенные библиотеки и компоненты, необходимые для запуска приложения на основе Qt. Загрузить инструмент можно по ссылке https://github.com/probonopd/linuxdeployqt/releases

Публикация для macOS

Для публикации приложения для MacOS Qt предоставляет инструмент macdeployqt, который аналогичен windeployqt и который можно найти в папке Qt в каталоге "bin". Так, этой утилите передается путь к приложению. Для приложения Qt Quick также передается параметр -qmldir с путем к файлам QML:

```
macdeployqt /Users/foo/myapp-build/MyApp.app -qmldir=/Users/foo/myapp/qml -dmg
```