Сборка QGIS с возможностью отладки в Visual C++ Express Edition 2008 или Qt Creator

Обсудить в форуме Комментариев — 4

Как скомпилировать QGIS под Windows и обеспечить возможность отладки

Одним из условий нормальной разработки программного обеспечения или модулей расширения является простота отладки. Ведь под отладчиком намного проще понять поведение программы и отловить возможные ошибки, т. к. аномальное поведение или логирование не всегда дает адекватное представление о поведении программы с теми или иными данными. Очень часто, когда возникает неправильное поведение программы, я прошу данные на которых это происходит и смотрю в отладчике на поведение программы.

В данной статье пойдет речь об отладке ГИС с открытым исходным кодом Quantum GIS (QGIS) в операционной системе Windows.

Содержание

Введение

Quantum GIS — свободная кроссплатформенная геоинформационная система, созданная с помощью инструментария Qt. Qt — кросс-платформенный инструментарий разработки ПО на языке программирования C++, отличительная особенность которого — использование Meta Object Compiler (MOC) — предварительной системы обработки исходного кода (в общем-то, Qt — это библиотека не для чистого C++, а для его особого диалекта, с которого и MOC и «переводит» код для последующей компиляции любым стандартным C++ компилятором). Утилита MOC ищет в заголовочных файлах на C++ описания классов, содержащие макрос Q_OBJECT, и создаёт дополнительный исходный файл на C++, содержащий мета-объектный код.

Для компиляции и отладки будем использовать только **свободные инструменты**: Microsoft Visual Studio Express 2008 и Qt Creator 2.1.

После изучения вопроса выяснилось практически полное отсутствие нормальной методики по компиляции и отладке под Windows (как я понял основная разработка ведется в ОС Linux). Хотя на официальном сайте в Wiki есть раздел Building on Windows, и описанная там методика (кстати рабочая) позволяет скомпилировать QGIS и даже запустить в режиме отладки. Но если вы внесете изменения в исходные коды, то вам придется заново через систему сборки CMake сгенерировать проекты под Visual Studio (из-за MOC). В случае полноценной Microsoft Visual Studio и специального дополнения QT Visual Studio Add-in такой проблемы нет, но это уже платный софт.

Подготовка

Первым делом скачаем и установим Visual Studio Express.

Для компиляции QGIS я решил использовать предварительно скомпилированные библиотеки от Qt под Windows для Visual Studio 2008 (забираем здесь). Это сэкономит время на компиляцию Qt (говорят эта процедура может продлиться более 5 часов). Не покидая сайт закачаем и Qt Creator.

Далее скачиваем и ставим Microsoft Windows Server 2003 R2 Platform SDK (для setupapi).

Кроме того, для сборки необходимы:

1. CMake
2. Flex (ставить только по путям без пробелов!)
3. Bison (ставить только по путям без пробелов!)

Скачиваем и устанавливаем.

Кроме того, для отладки в Qt Creator необходим CDB (из пакета MS Debugging Tools). Берем здесь.

Важно! После настройки отладчика в Qt Creator (Инструменты → Параметры → Отладчик → CDB) в какой-то момент он пропишет в поле «Пути к символам» путь к on-line БД символов Microsoft (начинается на http://). Рекомендуем удалить его, т. к. его присутствие вызывает неимоверные задержки при отладке.

На этом этапе у нас будет все необходимое программное обеспечение для сборки и отладки QGIS.

Следующим шагом идет загрузка и сборка библиотек от которых зависит QGIS.

Сборка библиотек

Мы будем компилировать и отлаживать QGIS в минимально возможной конфигурации. Для этого необходимы следующие библиотеки.

1. expat — библиотека нужна для работы с xml. В корне лежит expat.dsw который нормально открывается студией и компилируется. Я компилировал проект expatw (статически библиотека тоже скомпилировалась, но отказалась подключаться). Рекомендую установить переменную среды EXPAT на папку, куда вы распаковали expat.
2. zlib — библиотека нужна для упаковки/распаковки. В папке zlib-1.2.5\contrib\vstudio\vc9 найдется проект под студию. Рекомендую установить переменную среды ZLIB на папку, куда вы распаковали zlib.
3. iconv — библиотека нужна для преобразования кодировок строк. Для компиляции в Visual Studio пришлось действовать по инструкции отсюда. Рекомендую установить переменную среды ICONV на папку, куда вы распаковали iconv.
4. curl — библиотека для по протоколам Интернета (http, ftp и др.). Компилировал только поддержку HTTP без SSL. Рекомендую установить переменную среды CURL на папку, куда вы распаковали curl.
5. geos — библиотека для операций с геометрией. Рекомендую установить переменную среды GEOS на папку, куда вы распаковали geos.
6. proj4 — библиотека для операций с проекциями и преобразований систем координат. Рекомендую установить переменную среды PROJ на папку, куда вы распаковали proj4.
7. boost — библиотека расширяет стандартную библиотеку C++. Нужна если необходимо использовать поддержку KML в GDAL/OGR. Компиляция не требуется. Следует установить переменную среды BOOST_ROOT на папку, куда вы распаковали boost.
8. gdal — библиотека для работы с большим количеством растровых и векторных геопространственных форматов. Я закачал через TortoiseSVN альтернативный (свой) вариант отсюда. Единственное, там есть завязки на wxWidgets (wxzlib и wxexpat), которые надо поменять на соответствующие библиотеки zlib и expat.
9. qwt5 — библиотека виджетов для программирования приложений, имеющих техническую направленность. Собирал как написано в install.txt в секции под Windows — проблем не возникло.

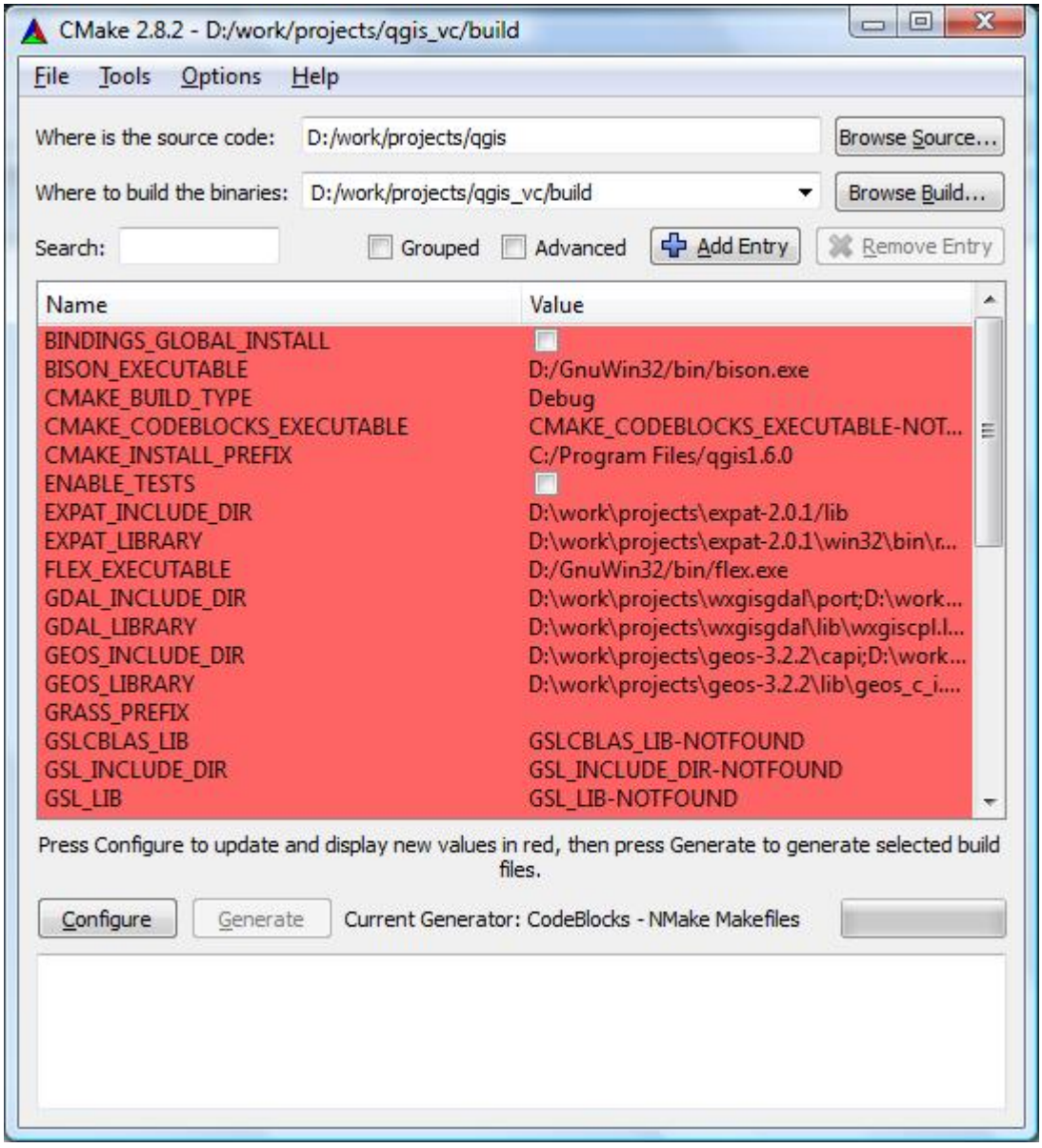
Все библиотеки я компилировал в Visual Studio Express 2008.

Подготовка к сборке QGIS

После компиляции всех вышеперечисленных библиотек у вас должны быть как минимум файлы *.lib, а еще (в основной массе) и *.dll. Кроме того, нужно знать где лежат еще и заголовочные файлы. Это понадобиться на этапе подготовки к сборке в CMake.

На данном этапе, настал момент для скачивания исходников QGIS. Я сделал это при помощи TortoiseSVN взяв их отсюда (https://svn.osgeo.org/qgis/trunk/qgis).

Для подготовки к сборке необходимо запустить CMake и указать путь к папке, где находятся исходники QGIS, а также путь в к папке, в которою запишется конфигурация (лучше указывать новую папку чтобы не смешивать исходные данные и сгенерированные).



При этом Cmake проверит конфигурацию и будет задавать вопросы о наличии тех или иных необходимых компонент (отмечено красным). Чтобы это произошло нажимаем «Configure» и указываем компилятор «NMake Makefiles» и оставляем отметку «Use default native compilers». Опишу свою конфигурацию.

```
BINDINGS_GLOBAL_INSTALL - нет
BISON_EXECUTABLE - D:/GnuWin32/bin/bison.exe
CMAKE_BUILD_TYPE - Debug
```

```
CMAKE_CODEBLOCKS_EXECUTABLE - CMAKE_CODEBLOCKS_EXECUTABLE-NOTFOUND
CMAKE_INSTALL_PREFIX - C:/Program Files/qgis1.6.0
ENABLE_TESTS - нет
EXPAT_INCLUDE_DIR - D:\work\projects\expat-2.0.1/lib
EXPAT_LIBRARY - D:\work\projects\expat-2.0.1\win32\bin\release\libexpat.lib
FLEX_EXECUTABLE - D:/GnuWin32/bin/flex.exe
GDAL_INCLUDE_DIR -
D:\work\projects\wxgisgdal\port;D:\work\projects\wxgisgdal\alg;D:\work\projects\wxgisgd
al\gcore;D:\work\projects\wxgisgdal\ogr
```

Важно! Если у вас несколько параметров (как у меня вместо одной библиотеки GDAL имеются 3 из wxGISGDAL), то параметры разделяются точкой с запятой.

```
GDAL_LIBRARY -
D:\work\projects\wxgisgdal\lib\wxgiscpl.lib;D:\work\projects\wxgisgdal\lib\wxgisogr.lib
;D:\work\projects\wxgisgdal\lib\wxgisgdal.lib
GEOS_INCLUDE_DIR - D:\work\projects\geos-3.2.2\capi;D:\work\projects\geos-
3.2.2\source\headers
GEOS_LIBRARY - D:\work\projects\geos-3.2.2\lib\geos_c_i.lib;D:\work\projects\geos-
3.2.2\lib\geos.lib
GRASS_PREFIX -
GSLCBLAS_LIB - GSLCBLAS_LIB-NOTFOUND
GSL_INCLUDE_DIR - GSL_INCLUDE_DIR-NOTFOUND
GSL_LIB - GSL_LIB-NOTFOUND
ICONV_INCLUDE_DIR - D:\work\projects\libiconv-1.13.1\include
ICONV_LIBRARY - D:\work\projects\libiconv-1.13.1\lib\iconv.lib
PEDANTIC - нет
POSTGRESQL_PREFIX -
PROJ_INCLUDE_DIR - D:\work\projects\proj-4.7.0\src
PROJ_LIBRARY - D:\work\projects\proj-4.7.0\wxGIS\lib\proj4.lib
QT_QMAKE_EXECUTABLE - D:/Qt/4.7.2/bin/qmake.exe
QWT_INCLUDE_DIR - D:/work/projects/qwt-5.2.1/src
QWT_LIBRARY - D:/work/projects/qwt-5.2.1/lib/qwt5.lib
SETUPAPI_LIBRARY - C:/Program Files/Microsoft Platform SDK for Windows Server 2003
R2/Lib/SetupAPI.Lib
SVN_MARKER - SVN_MARKER-NOTFOUND
WITH_BINDINGS - нет
WITH_GRASS - нет
WITH_INTERNAL_SPATIALITE - да
WITH_MAPSERVER - нет
WITH_POSTGRESQL - нет
WITH_SPATIALITE - нет
```

Чтобы не сбрасывать файл unistd.h из состава GnuWin32 я в файле CMakeLists.txt внес такую строчку:

```
SET(UNISTD_INCLUDE_DIR "D:\GnuWin32\include")
```

Кроме того в файлы D:\work\projects\qgis\src\analysis\CMakeLists.txt и D:\work\projects\qgis\src\core\CMakeLists.txt добавил в конец раздела

```
INCLUDE_DIRECTORIES(
  ${CMAKE_CURRENT_SOURCE_DIR}
  ...
  ${UNISTD_INCLUDE_DIR}
)
```

Configure должно проходить без ошибок. Далее шаги для Visual Studio Express и Qt Creator различаются.

MS Visual Studio Express

В Visual Studio Express создаем новый проект типа Makefile и ничего не настраивая жмем Finish.

Далее заходим в свойства созданного проекта и в разделе NMake пишем (учитывая ваши пути естественно):

В подразделе **Build Command Line**

```
set BUILD_DIR=D:\work\projects\qgis_vc\build
set PROJECT_DIR=D:\work\projects\qgis_vc\build\debug
cmake -E make_directory %BUILD_DIR%
cd %BUILD_DIR%
cmake -G «NMake Makefiles» %PROJECT_DIR%
nmake all
```

```
set BUILD_DIR=D:\work\projects\qgis_vc\buildset
PROJECT_DIR=D:\work\projects\qgis_vc\build\debug
cmake -E make_directory %BUILD_DIR%cd %BUILD_DIR%cmake -G «NMake
Makefiles» %PROJECT_DIR%nmake all
```

В подразделе **Rebuild All Command Line**

```
set BUILD_DIR=D:\work\projects\qgis_vc\build
set PROJECT_DIR=D:\work\projects\qgis_vc\build\debug
cmake -E make_directory %BUILD_DIR%
cd %BUILD_DIR%
cmake -G «NMake Makefiles» %PROJECT_DIR%
nmake clean all
```

В подразделе **Clean Command Line**

```
set BUILD_DIR=D:\work\projects\qgis_vc\build
set PROJECT_DIR=D:\work\projects\qgis_vc\build\debug
cmake -E make_directory %BUILD_DIR%
cd %BUILD_DIR%
cmake -G «NMake Makefiles» %PROJECT_DIR%
nmake clean
```

В подразделе **Output**

```
qgis.exe
```

Далее в проект добавляем исходные тексты для установок точек останова (в принципе, проект и без исходников скомпилируется, но отлаживать его не получится). Жмем F7 и смотрим ошибки компиляции. Если ошибок нет, необходимо сбросить в созданную папку %BUILD_DIR%, куда скомилировались наши файлы QGIS, связанные библиотеки. У меня их состав такой:

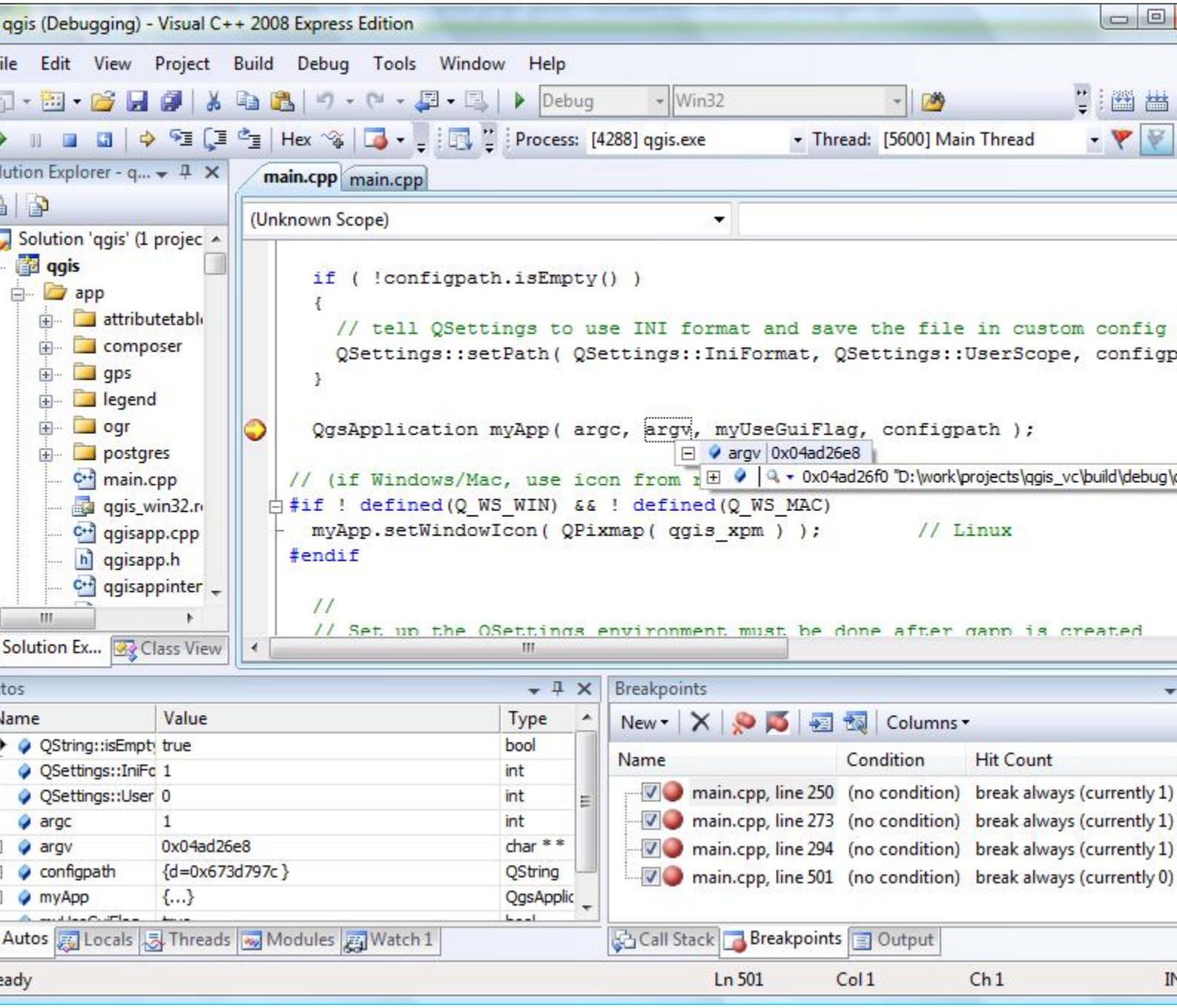
- geos_c.dll
- iconv.dll
- libcurl.dll
- **libeay32.dll**
- libexpat.dll
- **libiconv-2.dll**

- **libintl-8.dll**
- **libpq.dll**
- msvcm90d.dll
- msvcp90d.dll
- msvcr90.dll
- msvcr90d.dll
- proj4.dll
- qwt5.dll
- ssleay32.dll
- wxgiscpl.dll
- wxgisgdal.dll
- wxgisogr.dll
- zlibwapi.dll

Жирным выделены библиотеки, которые необходимы в случае если GDAL скомпилирован с поддержкой PostGIS.

На этом этапе QGIS должен запускаться, для этого жмем F5. Если запустится, то можно отлаживать. Кроме того, при запуске будет некоторая ругань на отсутствующие директории, которые можно взять из установки QGIS. Также я создал папку %BUILD_DIR%/plugins куда перенес скомпилированные плагины QGIS.

Для отладки ставим точку останова и жмем опять же F5. У меня получилось следующее

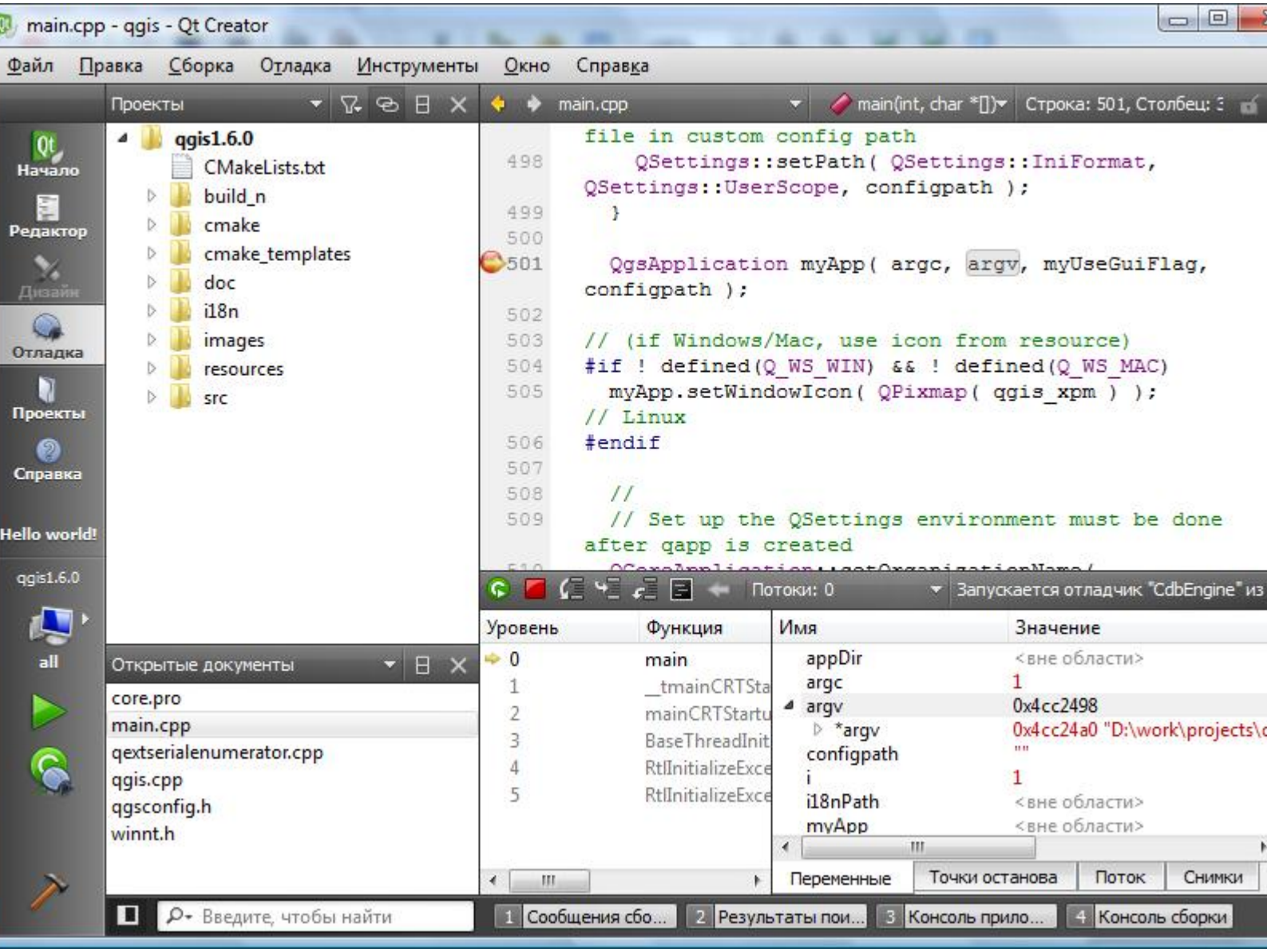


Qt Creator

Для отладки в Qt Creator делаем следующее.

Выбираем «Открыть файл или проект» и выбираем файл CMakeListst.txt, а в качестве выходной директории указываем туже, что и в CMake. Далее компилируем и исправляем ошибки компиляции, если они есть.

Если ошибок нет, необходимо сбросить в созданную папку, куда скомилировались наши файлы QGIS, связанные библиотеки (состав библиотек см. выше). Также создаем папку plugins и переносим скомпилированные плагины QGIS. Если QGIS запускается можно отлаживать. Ставим точку останова и жмем F5. Должно получиться следующее



[Обсудить в форуме](#) Комментариев — 4

Последнее обновление: 2014-05-15 01:35

Дата создания: 02.04.2011
Автор(ы): [Дмитрий Барышников](#)