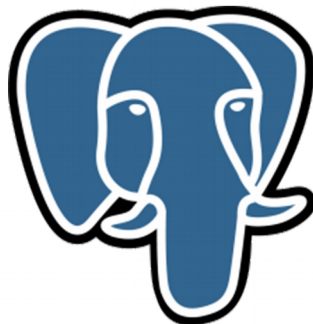


Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Тихоокеанский государственный университет»

Тимошко А.М., Тусикова А.А.

# Сборка Qt5 без прав администратора с поддержкой драйвера PostgreSQL

Методическое пособие к лабораторной работе №2 по дисциплине  
«Проектирование приложений баз данных» для студентов  
специальности 231000.62 «Программная инженерия»



Хабаровск  
2016

Сборка Qt5 без прав администратора с поддержкой драйвера PostgreSQL: методическое пособие к лабораторной работе №2 по дисциплине «Проектирование приложений баз данных» для студентов специальности 231000.62 «Программная инженерия» / сост. Тимошко А.А., Тусикова А.А. – Хабаровск: 2016. – 19 с.

Методическое пособие к лабораторной работе №2 по дисциплине «Проектирование приложений баз данных» составлены студентами для помощи другим студентам при выполнении данной работы. В них изложен доступным языком материал для практического применения: сборка и установка Qt5 с поддержкой драйвера PostgreSQL двумя способами, создание проекта в Qt5 для подключения к СУБД PostgreSQL.

Хабаровск, 2016

## **Оглавление**

Шаг 1. Сборка и установка Qt5	4
Шаг 2. Создание тестового приложения для проверки работы с СУБД PostgreSQL.	7
Шаг 3. Установка справочной системы.	9
Используемые источники	10
Приложение А. Ключи конфигурирования Qt5	11

Для данной работы будем собирать Qt версии 5.10.1 с поддержкой драйвера PostgreSQL 9.3. Рассмотрим два варианта:

- встраивание драйвера `psql` как плагин на этапе сборки Qt5.
- компиляция и встраивание драйвера `psql` как плагин после установки Qt5 вручную.

## Шаг 1. Сборка и установка Qt5

1. Для начала скачайте исходные файлы Qt 5.5.1 по ссылке [\[1\]](#) и распакуйте в отдельную папку. Большинство необходимых для функционирования Qt библиотек, как правило, уже установлены в системе, имеющий графический рабочий стол. Если при сборке Qt, всё таки возникли проблемы, связанные с отсутствием необходимых библиотек, то их можно решить установкой следующих библиотек (на примере ОС Ubuntu):

- `libfontconfig1-dev`
- `libfreetype6-dev`
- `libx11-dev`
- `libxext-dev`
- `libxfixes-dev`
- `libxi-dev`
- `libxrender-dev`
- `libxcb1-dev`
- `libx11-xcb-dev`
- `libxcb-glx0-dev`

Команда для установки:

```
sudo apt-get install <name-dev1> <name-dev2> ... <name-devN>
```

2. Создайте папку в домашней директории, в которую будет установлен Qt5. Избегайте кириллицы и пробелов в пути к данной папке.
3. Перейдите в папку с исходными файлами Qt5. В зависимости от варианта установки Qt5 с поддержкой драйвера PostgreSQL параметры автоконфигурации пакета и дальнейшие действия различаются.

## ***Вариант 1. Сборка Qt5 с поддержкой PostgreSQL.***

1. Перейдите в папку с исходными текстами Qt.
2. `./configure -prefix /home/user/Qt5.10.1 -qt-xcb -system-freetype -fontconfig -nomake examples -nomake tests -skip qtwebengine -opensource -confirm-license -I/home/user/POSTGRESQL/include -L/home/user/POSTGRESQL/lib -plugin-sql-psql`

Рассмотрим вкратце используемые здесь ключи конфигурации:

- `qt-xcb` – реализация при сборке в Qt аналога библиотеки `xcb`. Данная библиотека позволяет взаимодействовать С-программам с X-сервером. Обязательна для GUI-приложений для ОС семейства Linux;
- `system-freetype` — использование системных шрифтов (обязателен пакет `libfreetype6-dev`);
- `fontconfig` — поддержка сглаживания шрифтов. Без данного пакета шрифты в приложениях выглядят “угловато” (обязателен пакет `libfontconfig1-dev`). В последних версиях библиотеки Qt5 реализовано сглаживание шрифта без внешних библиотек. Поэтому необходимость в данном ключа отпала;
- `nomake` — отключение некоторой задачи при сборке. В приведённом выше примере отключена сборка примеров (`examples`) и проведение тестирования сборки (`tests`);
- `skip` — пропуск сборки некоторого модуля. В приведённом выше примере отключена сборка модуля `qtwebengine`. Данный модуль используется для создания web-приложений;
- `opensource` — автоматический выбор лицензии GPL. Без данного ключа это придётся делать вручную;
- `confirm-license` — автоматическое подтверждение лицензии GPL. Без данного ключа это придётся делать вручную;
- `-I` и `-L` — пути к заголовочным и двоичным файлам программного продукта (в данном случае СУБД PostgreSQL);

- `plugin-sql-psql` — сборка драйвера как двоичной динамической библиотеки.

Процесс конфигурирования занимает примерно 3-5 мин (ЦП Intel Core i3 1.8 GHz, ОЗУ 4 Гб). Все возможные ключи для конфигурирования Qt5 описаны в приложении А.

3. Выполните команду **make**. Данный процесс довольно длительный и занимает от 30 мин до нескольких часов, в зависимости от конфигурации компьютера (ЦП Intel Core i3 1.8 GHz, ОЗУ 4 Гб – 4,5 ч). Для ускорения сборки можно использовать ключ `-j<n>` утилиты `make`. Здесь `n` — целое число. Как правило, для определения `n` используют формулу:

`n=<число ядер CPU>+1`.

Ключ `j` заставляет утилиту `make` выполнять поставленную задачу в несколько потоков. Грамотное использование ключей `nomake` и `skip` также ускоряет сборку библиотеки.

4. Выполните команду **make install**.
5. Qt-драйвер PostgreSQL бóльшую часть своей работы выполняет путём вызова функций другой библиотеки: `libpq.so`. Чтобы обеспечить нормальное функционирование Qt-драйвера при работе приложений, необходимо установить переменную окружения `LD_LIBRARY_PATH`. Сделать это можно двумя способами:

1) Поместить в конец файла `~/.bashrc` строчку:

```
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/home/user/
POSTGRESQL/lib/
```

2) Для созданного проекта в Qt Creator настроить среду запуска:

«Проекты – Среда запуска – Подробнее – Добавить».

Переменная `LD_LIBRARY_PATH`, путь

```
/home/user/POSTGRESQL/lib/
```

На этом сборка и установка Qt с поддержкой драйвера PostgreSQL закончена.

**Вариант 2. Сборка Qt5 без поддержки драйвера PostgreSQL с дальнейшим его подключением в качестве плагина.**

1. `./configure -prefix /home/user/Qt5.10.1 -qt-xcb -system-freetype -fontconfig -nomake examples -nomake tests -skip qtwebengine -opensource -confirm-license`
2. `make`
3. `make install`
4. После установки Qt5 перейдите в папку с исходными файлами Qt5 для сборки драйвера:  
`cd /home/user/Qt5.10.1-src/qtbase/src/plugins/sqldrivers/psql`
5. Соберите проект psql.pro (файл для сборки драйвера) при помощи qmake с указанием путей к заголовочным файлам и библиотекам установленной СУБД PostgreSQL:  
`/home/user/Qt5.10.1/bin/qmake`  
`"INCLUDEPATH+=/home/user/POSTGRESQL/include"`  
`"LIBS+=-L/home/user/POSTGRESQL/lib -lpq" psql.pro`
6. `make install`  
Если все прошло успешно, в папке `/home/user/Qt5.10.1/plugins/sqldrivers` должен появиться файл `libqsqlpsql.so`.
7. Для нормального функционирования драйвера, необходимо выполнить точно такие же действия, что и в пятом пункте первого варианта;  
На этом сборка и установка Qt с последующей установкой поддержки драйвера PostgreSQL закончена.

## **Шаг 2. Создание тестового приложения для проверки работы с СУБД PostgreSQL.**

1. В отдельной папке создайте два файла: `test.pro` и `main.cpp`.
2. Содержимое файла `test.pro`:

```
SOURCES += \  
main.cpp  
TARGET = test  
QT = core sql  
  
CONFIG += console
```

3. Содержимое файла `main.cpp`:

```

#include <QtCore>
#include <QtSql>
#include <QDebug>

int main()
{
    QSqlDatabase db;

    // Добавляем БД на PostgreSQL в список баз
    db = QSqlDatabase::addDatabase("QPSQL");

    // устанавливаем имя узла сервера БД
    db.setHostName("10.10.138.35");

    // устанавливаем порт соединения
    db.setPort(5432);

    // устанавливаем имя БД для соединения
    db.setDatabaseName("database");

    // устанавливаем имя пользователя для соединения
    db.setUserName("user");

    // устанавливаем пароль пользователя
    db.setPassword("12e89s46l3g");

    // пытаемся открыть соединение с БД
    if(!db.open())
    {
        // в случае неудачи - вывести текст ошибки
        qDebug()<<"Error: "<<db.lastError().databaseText());
        return 0;
    }
    // в случае успеха – создание тестовой таблицы
    db.exec("create table t(a integer)");

    return 0;
}

```

4. Выполните в текущей папке следующие команды:  
[/home/user/Qt5.10.1/bin/qmake test.pro](#)  
[make](#)
5. В случае отсутствия ошибок компиляции, в текущей папке появится исполняемый файл с именем test. Его можно запустить на выполнение при помощи команды:  
[./test](#)



6. В случае успешной компиляции драйвера и успешного подключения к серверу PostgreSQL по адресу “10.10.138.35” в БД с именем “database” появится таблица с именем “t”, состоящая из единственного поля с именем “a” целого типа. Это означает успешное выполнение работы.

### **Шаг 3. Установка справочной системы.**

Большим достоинством Qt является наличие хорошей документации. Она присутствует и в исходных текстах, но только в “сыром” (неподготовленном) виде. Наиболее популярным форматом файлов справочной системы Qt является qch. Именно он используется для предоставления справочной информации в программе QtCreator. Для установки справки в формате qch, необходимо выполнить следующие шаги:

1. Перейдите в папку с исходными текстами Qt:  
`cd /home/user/Qt5.10.1-src`
2. Выполните в этой папке две команды:  
`make docs`  
`make install_docs`
3. После завершения процесса построения и установки справки в папке doc установки Qt (в нашем случае /home/user/Qt5.10.1/doc) появится набор файлов справочной системы с расширением qch. При интеграции данной сборки Qt в программу QtCreator вся справочная системы станет доступной для пользователя.

## **Используемые источники**

1. <https://www1.qt.io/offline-installers/>
2. <http://doc.qt.io/qt-5/linux-requirements.html>
3. <http://doc.qt.io/qt-5/configure-options.html>
4. <http://qt.developpez.com/doc/5.0-snapshot/configure-options/>
5. <http://doc.qt.io/qt-5/linux-building.html>
6. <http://doc.qt.io/qt-5/sql-driver.html>
7. <https://habrahabr.ru/post/173283/>

## Приложение А. Ключи конфигурирования Qt5

Ключ	Описание
<b>Варианты установки</b>	
-prefix <dir>	Каталог установки на целевом устройстве (default /usr/local/Qt-5.5.1)
-extprefix <dir>	Каталог установки на хост-машине (default SYSROOT/PREFIX)
-hostprefix [dir]	Каталог установки инструментов для сборки, запускаемых на хост-машине (default EXTPREFIX)
<i>Все каталоги, кроме –sysconfdir, должны быть расположены под -prefix / -hostprefix</i>	
-bindir <dir>	Каталог установки исполняемых файлов пользователя (default PREFIX/bin)
-headerdir <dir>	Каталог установки заголовочных файлов (default PREFIX/include)
-libdir <dir>	Каталог установки библиотечных файлов (default PREFIX/lib)
-archdatadir <dir>	Каталог установки архитектурно-зависимых данных, используемых Qt (default PREFIX)
-plugindir <dir>	Каталог установки плагинов (default ARCHDATADIR/plugins)
-libexecdir <dir>	Каталог установки программных исполняемых файлов (default ARCHDATADIR/libexec, ARCHDATADIR/bin for MinGW)
-importdir <dir>	Каталог импорта QML1 (default ARCHDATADIR/imports)
-qmldir <dir>	Каталог импорта QML2 (default ARCHDATADIR/qml)
-datadir <dir>	Каталог установки архитектурно-зависимых данных, используемых Qt (default PREFIX)
-docdir <dir>	Каталог установки документации (default DATADIR/doc)

-translationdir <dir>	Каталог установки языков Qt (default DATADIR/translations)
-sysconfdir <dir>	Каталог установки параметров Qt (default PREFIX/etc/xdg)
-examplesdir <dir>	Каталог установки примеров (default PREFIX/examples)
-testsdir <dir>	Каталог установки тестов (default PREFIX/tests)
-hostbindir <dir>	Каталог установки исполняемых файлов хоста (default HOSTPREFIX/bin)
-hostlibdir <dir>	Каталог установки библиотечных файлов хоста (default HOSTPREFIX/lib)
-hostdatadir <dir>	Каталог установки данных, используемых qmake (default HOSTPREFIX)
<b>Настройка параметров</b>	
<i>(*) обозначены значения по умолчанию, (+) обозначены значения по умолчанию, требующих оценку (если оценка положительна, то значение используется)</i>	
* -release	Скомпилировать и слинковать Qt с отключённой отладкой
-debug	Скомпилировать и слинковать Qt с включённой отладкой
-debug-and-release	Скомпилировать и слинковать две библиотеки Qt - с включённой и выключенной отладкой (только для Mac OS)
-force-debug-info	Создать символьные файлы для релизной сборки
-developer-build	Скомпилировать и слинковать Qt с инструментами разработчика
-opensource	Скомпилировать и слинковать Qt с открытыми исходными кодами
-commercial	Скомпилировать и слинковать коммерческую редакцию Qt
-confirm-license	Автоматически принять лицензию
-no-c++11	Скомпилировать Qt без поддержки C++11
+ -c++11	Скомпилировать Qt с поддержкой C++11
* -shared	Создать и использовать разделяемые библиотеки Qt
-static	Создать и использовать статические библиотеки

-no-largefile	Отключить поддержку больших файлов
+ -largefile	Включить поддержку файлов больше 4 Гб
-no-accessibility	Не компилировать поддержку Accessibility
+ -accessibility	Компилировать поддержку Accessibility
-no-sql-<driver>	Отключить драйвер полностью
-qt-sql-<driver>	Включить драйвер в библиотеке Qt SQL
-plugin-sql-<driver>	Включить драйвер как плагин во время выполнения
-system-sqlite	Использовать SQLite из ОС
-no-qml-debug	Не компилировать поддержку QML-отладки
+ -qml-debug	Компилировать поддержку QML-отладки
-platform <target>	ОС и компилятор, с которым делается сборка
-no-sse2	Не компилировать с использованием SSE2
-no-sse3	Не компилировать с использованием SSE3
-no-sse4.1	Не компилировать с использованием SSE4.1
-no-sse4.2	Не компилировать с использованием SSE4.2
-no-avx	Не компилировать с использованием AVX
-no-avx2	Не компилировать с использованием AVX2
-no-mips_dsp	Не компилировать с использованием MIPS DSP
-no-mips_dspr2	Не компилировать с использованием MIPS DSP rev 2
-qtnamespace <name>	Обернуть весь код библиотек Qt в пространстве имен 'namespace <name> {...}'
-qtlibinfix <infix>	Переименовать все libQt*.so в libQt*<infix>.so
-testcocoona	Инструмент TestCocoona
-gcov	Инструмент GCov
-D <string>	Добавить явно директиву препроцессора
-I <string>	Добавить явно путь к заголовочным файлам
-L <string>	Добавить явно путь к библиотечным файлам
+ -pkg-config	Использовать pkg-конфигурацию для обнаружения путей к заголовочным и библиотечным файлам
-no-pkg-config	Не использовать pkg-конфигурацию
-force-pkg-config	Принудительно использовать pkg-config
-help, -h	Показать информацию о ключах configure
<b>Сторонние библиотеки</b>	
-qt-zlib	Использовать zlib, поставляемый с Qt
+ -system-zlib	Использовать zlib, установленный в системе
-no-mtdev	Не компилировать поддержку mtdev

+ -mtdev	Включить поддержку mtdev
+ -no-journald	Не отправлять вход / выход в journald
-journald	Отправлять вход / выход в journald
-no-gif	Не компилировать поддержку GIF
-no-libpng	Не компилировать поддержку PNG
-qt-libpng	Использовать libpng, поставляемый с Qt
+ -system-libpng	Использовать libpng, установленный в системе
-no-libjpeg	Не компилировать поддержку JPEG
-qt-libjpeg	Использовать libjpeg, поставляемый с Qt
+ -system-libjpeg	Использовать libjpeg, установленный в системе
-no-freetype	Не компилировать поддержку Freetype2
-qt-freetype	Использовать freetype, поставляемый с Qt
+ -system-freetype	Использовать freetype, установленный в системе
-no-harfbuzz	Не компилировать поддержку HarfBuzz-NG
* -qt-harfbuzz	Использовать harfbuzz, поставляемый с Qt
-system-harfbuzz	Использовать harfbuzz, установленный в системе
-no-openssl	Не компилировать поддержку OpenSSL
+ -openssl	Включить поддержку OpenSSL во время выполнения
-openssl-linked	Включить линковку OpenSSL
-no-libproxy	Не компилировать поддержку libproxy
+ -libproxy	Использовать libproxy, установленный в системе
-qt-pcre	Использовать PCRE , поставляемый с Qt
+ -system-pcre	Использовать PCRE, установленный в системе
-qt-xcb	Использовать xcb, поставляемый с Qt
+ -system-xcb	Использовать xcb, установленный в системе
-xkb-config-root	Установить корень XKB-конфигурации по умолчанию (используется только с опцией -qt-xkbcommon-x11)
-qt-xkbcommon-x11	Использовать xkbcommon, поставляемый с Qt в сочетании с xcb
+ -system-xkbcommon-x11	Использовать libproxy, установленный в системе в сочетании с xcb
-no-xkbcommon-evdev	Не использовать X-less xkbcommon при компиляции libinput
* -xkbcommon-evdev	Использовать X-less xkbcommon при компиляции libinput
-no-xinput2	Не компилировать поддержку XInput2

* -xinput2	Компилировать поддержку XInput2
-no-xcb-xlib	Не компилировать поддержку Xcb-Xlib
* -xcb-xlib	Компилировать поддержку Xcb-Xlib
-no-glib	Не компилировать поддержку Glib
+ -glib	Компилировать поддержку Glib
-no-pulseaudio	Не компилировать поддержку PulseAudio
+ -pulseaudio	Компилировать поддержку PulseAudio
-no-alsa	Не компилировать поддержку ALSA
+ -alsa	Компилировать поддержку ALSA
-no-gtkstyle	Не компилировать поддержку GTK
+ -gtkstyle	Компилировать поддержку GTK
<b><i>Дополнительные параметры</i></b>	
-make <part>	Добавить список компонентов, которые будут собраны при выполнении make
-nomake <part>	Добавить список компонентов, которые не будут собраны при выполнении make
-skip <module>	Исключить весь модуль из сборки
-no-compile-examples	Установить только исходные тексты примеров без компиляции
-no-gui	Не компилировать Qt GUI и зависимости
+ -gui	Компилировать Qt GUI и зависимости
-no-widgets	Не компилировать модуль Qt Widgets
+ -widgets	Компилировать модуль Qt Widgets
-R <string>	Добавить явный путь к Qt библиотекам во время выполнения
-l <string>	Добавить явно библиотеку
-no-rpath	Не использовать путь к библиотеке как путь к библиотеке во время выполнения
+ -rpath	Использовать путь к библиотеке как путь к библиотеке во время выполнения
-continue	Продолжить в случае возникновения предупреждений и ошибок
-verbose, -v	Выводить подробную информацию о каждом шаге
-silent	Снизить выходы из сборки, чтобы легче увидеть предупреждения и ошибки
* -no-optimized-qmake	Не собирать оптимизированный qmake
-optimized-qmake	Собирать оптимизированный qmake
-no-nis	Не компилировать поддержку NIS
* -nis	Компилировать поддержку NIS

-no-cups	Не компилировать поддержку CUPS
* -cups	Компилировать поддержку CUPS
-no-iconv	Не компилировать поддержку iconv
* -iconv	Компилировать поддержку iconv
-no-evdev	Не компилировать поддержку evdev
* -evdev	Компилировать поддержку evdev
-no-tslib	Не компилировать поддержку tslib
* -tslib	Компилировать поддержку tslib
-no-icu	Не компилировать поддержку ICU
+ -icu	Компилировать поддержку ICU
-no-fontconfig	Не компилировать поддержку FontConfig
+ -fontconfig	Компилировать поддержку FontConfig
-no-strip	Не удалять исполняемые файлы и библиотеки ненужных символов
* -strip	Удалять исполняемые файлы и библиотеки ненужных символов
* -no-pch	Не использовать прекомпиляцию заголовочных файлов
-pch	Использовать прекомпиляцию заголовочных файлов
-no-dbus	Не компилировать модуль Qt D-Bus
+ -dbus	Компилировать модуль Qt D-Bus и динамически загружать libdbus-1
-dbus-linked	Компилировать модуль Qt D-Bus и линковать libdbus-1
-reduce-relocations	Уменьшить передислоцирование в библиотеках через оптимизации линковщика
-no-use-gold-linker	Не линковать, используя GNU gold linker
+ -use-gold-linker	При возможности линковать, используя GNU gold linker
-force-asserts	Принудительно включить Q_ASSERT, даже в релизной версии
-sanitize [address thread memory undefined]	Включить указанный компилятор дезинфекции
-device <name>	Кросс-компиляция для устройства
-device-option <key=value>	Добавить специфичные опции устройства в mkspec
* -no-separate-debug-info	Не хранить отладочную информацию в отдельном файле



-separate-debug-info	Сохранять отладочную информацию в отдельный файл
-no-xcb	Не компилировать поддержку Xcb
* -xcb	Компилировать поддержку Xcb
-no-eglfs	Не компилировать поддержку EGLFS
* -eglfs	Компилировать поддержку EGLFS
-no-directfb	Не компилировать поддержку DirectFB
* -directfb	Компилировать поддержку DirectFB
-no-linuxfb	Не компилировать поддержку Linux Framebuffer
* -linuxfb	Компилировать поддержку Linux Framebuffer
-no-kms	Не компилировать поддержку KMS
* -kms	Компилировать поддержку KMS
-qpa <name>	Установить платформу QPA по умолчанию
-xplatform target	Целевая платформа при кросс-компиляции
-sysroot <dir>	Установить каталог для компилятора и qmake, а также путь PKg-конфигурации
-no-gcc-sysroot	При использовании -sysroot отключает -sysroot в компиляторе
-no-feature-<feature>	Не компилировать в <feature>
-feature-<feature>	Компилировать в <feature>
-qconfig local	Использовать src/corelib/global/qconfig-local.h, а не по умолчанию (полный)
-qreal [double float]	Использовать qreal как специальный тип. По умолчанию double
-no-opengl	Отключить поддержку OpenGL
-opengl <api>	Включить поддержку OpenGL
-no-libinput	Отключить поддержку libinput
* -libinput	Включить поддержку libinput
-no-gstreamer	Отключить поддержку GStreamer
+ -gstreamer <version>	Включить поддержку GStreamer
* -no-system-proxies	Не использовать системные сетевые прокси по умолчанию
-system-proxies	Использовать системные сетевые прокси по умолчанию
-no-warnings-are-errors	Предупреждения рассматриваются не как ошибки
-warnings-are-errors	Предупреждения рассматриваются как ошибки