# Trade History.

# Описание

Скрипт на LUA для расчета PnL по ФИФО, отображения текущих и закрытых позиций.

Версия 1.0

Для работы скрипт использует библиотеки sqlite3.dll (https://sqlite.org/download.html) и lsqlite3.dll ()

Адрес в интернете: <https://github.com/9159340/TradeHistory>

# Самостоятельная компиляция библиотек

Библиотеки уже скомпилированы и готовы к работе, но если вы желаете быть уверенными, что там нет никаких закладок и бэкдоров, то можете скомпилировать их сами. Инструкция приведена ниже.

**Компиляция Lsqlite3.dll**

Чтобы скомпилировать библиотеку lsqlite3.dll нужно выполнить следующие шаги

**Установить Компилятор MinGW**

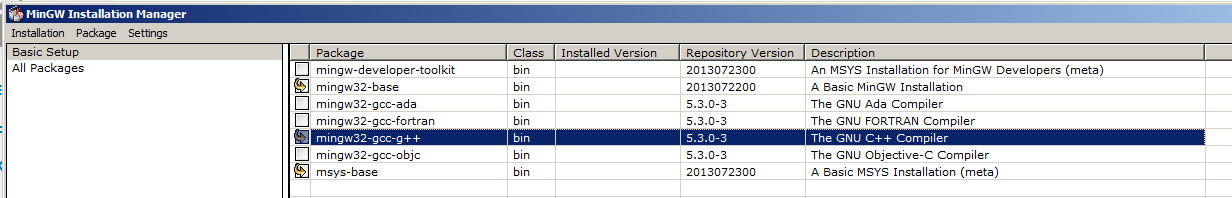
<http://www.mingw.org/wiki/Getting_Started>

<https://sourceforge.net/projects/mingw/files/>

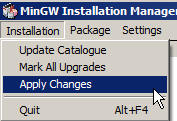
Прямая ссылка

<https://sourceforge.net/projects/mingw/files/latest/download?source=files>

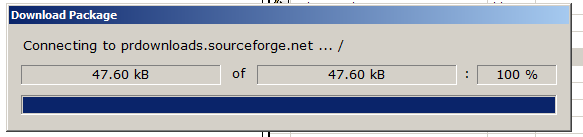
Скачиваем и устанавливаем инсталлятор. Запускаем. Включаем пакеты:



Применяем



Ждем завершения



После завершения установки нужно добавить путь c:\MinGW\bin\ в переменную Path (укажите свой путь к папке bin)

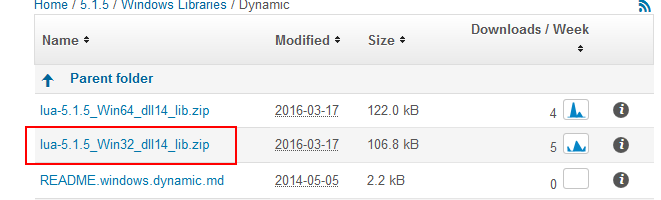
**Lua**

<http://luabinaries.sourceforge.net/download.html>

Нужно использовать версию 5.1

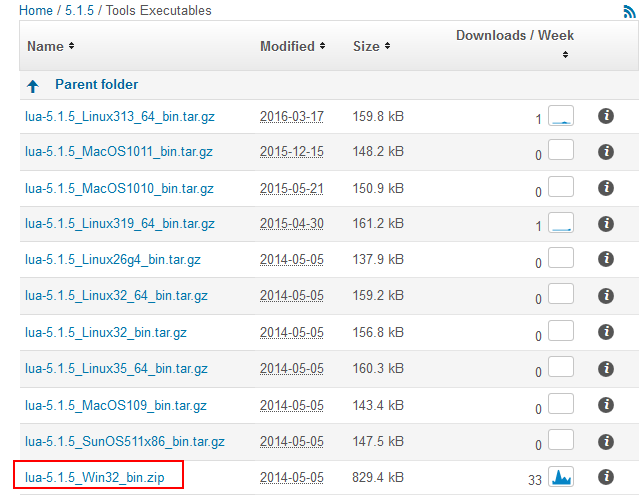
dll

<https://sourceforge.net/projects/luabinaries/files/5.1.5/Windows%20Libraries/Dynamic/>

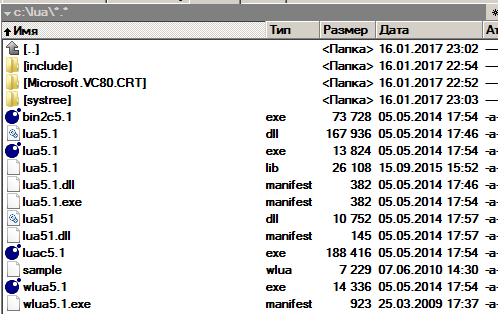


exe

<https://sourceforge.net/projects/luabinaries/files/5.1.5/Tools%20Executables/>



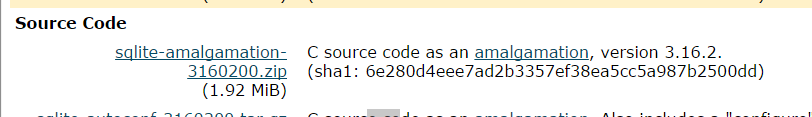
Архив lua-5.1.5\_Win32\_bin.zip распакуйте в папку, например, c:\lua, и добавьте туда папку include из архива lua-5.1.5\_Win32\_dll14\_lib.zip, чтобы получилось, как на скриншоте ниже



**sqlite3**

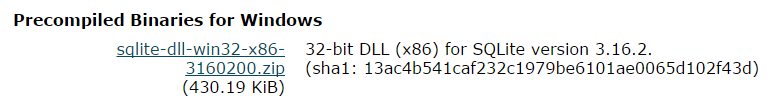
<https://sqlite.org/download.html>

Качаем файл заголовочные файлы (они нужны для установки lsqlite3 через luarocks)



содержимое архива раскладываем вручную, файл sqlite3.h кладем в папку c:\sqlite\include.

качаем dll



sqlite3.dll кладем в папку c:\sqlite\lib

**Luarocks**

Скачать и установить luarocks с параметром /MW для поддержки компилятора MinGW.

<http://luarocks.github.io/luarocks/releases/>

install.bat запускать от имени администратора.

Устанавливается он сюда:

c:\Program Files (x86)\LuaRocks\

Этот путь нужно прописать в переменной PATH

После всего нужно воспользоваться советом из <https://forum.quik.ru/forum10/topic1236/>, удалить строку MSVCRT = 'MSVCR80' из конфигурационного файла, иначе при запуске скрипта будет выдана ошибка «error loading module 'lsqlite3' from file '.\lsqlite3.dll': Не найден указанный модуль.»

**Lsqlite3**

<http://lua.sqlite.org/index.cgi/doc/tip/doc/lsqlite3.wiki>

<https://luarocks.org/modules/dougcurrie/lsqlite3>

Устанавливаем через luarocks. В командной строке пишем:

luarocks install sqlite3 SQLITE\_DIR=c:\sqlite

Файл библиотеки будет создан по адресу:

c:\lua\systree\lib\lua\5.3\lsqlite3.dll

Если имя файла будет вроде этого lsqlite3\_0\_9\_4\_2-lsqlite3.dll, то нужно его переименовать в lsqlite3.dll.

Оба файла, lsqlite3.dll и sqlite3.dll, нужно положить в каталог QUIK.

# Таблицы файла positions2.db

В базе данных находятся следующие таблицы:

|  |  |
| --- | --- |
| Имя таблицы | Назначение |
| deals | Хранение всех полей из сделки, которая приходит в событии OnTrade. Так как при совершении сделки возникает 2 события, то перед добавлением сделки в базу выполняется поиск по дате и номеру, чтобы не создавать дубли. |
| fifo\_long\_2 | «длинные» позиции, рассчитываемые по фифо |
| fifo\_short\_2 | «короткие» позиции, рассчитываемые по фифо |
| orders | Заявки, пока не используется |
| securities | Ценные бумаги. Там лежат коды фьючерсов и некоторые их параметры, используемые для расчета прибыли в рублях (шаг цены, стоимость шага цены, размер лота) |

Индексы таблиц

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица | Имя индекса | Поля и сортировка (ASC – по возрастанию, DESC – по убыванию) | Описание |
| deals | Num\_deals\_idx | Trade\_num ASC, date ASC | Используется при поиске сделки при добавлении новой в событии OnTrade |
| Fifo\_long\_2 | getRests | dim\_client\_code ASC, dim\_sec\_code ASC, dim\_class\_code ASC, dim\_brokerref ASC | Используется при получении текущих остатков позиций |
| Fifo\_short\_2 | getRests\_s | dim\_client\_code ASC, dim\_sec\_code ASC, dim\_class\_code ASC, dim\_brokerref ASC | Используется при получении текущих остатков позиций |
| Fifo\_long\_2 | sec\_date\_close | dim\_sec\_code ASC, close\_date ASC, close\_time ASC | Пока не используется, планировалось для отбора закрытых позиций |
| Fifo\_short\_2 | sec\_date\_cl\_short | dim\_sec\_code ASC, close\_date ASC, close\_time ASC | Пока не используется, планировалось для отбора закрытых позиций |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Контекстное меню в главной таблице робота.

Для более удобного закрытия/переворота позиций, а также других действий над позицией, которые можно запрограммировать, в главную таблицу добавлено контекстное меню. Т.к. для произвольных таблиц QUIK контекстное меню не предусмотрено, то в качестве воркэраунда будем показывать таблицу Lua.