

StrategyQuant

StrategyQuant X

User's Guide

Support

If you'll have trouble understanding anything, you need help, or you simply have some question to ask (related to the system), remember your purchase includes also a support.

We are here for you, you can contact us using our forum:

www.StrategyQuant.com/forum

Articles section on our website provide additional source of knowledge

docs.StrategyQuant.com

Copyright

All rights reserved.

StrategyQuant software, bonus strategies and content of this Manual is copyrighted.
You can use them only with valid license.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means - including electronic, mechanical, photocopy, recording, scanning or otherwise - without the prior written permission of the author.

© 2017-2018 StrategyQuant s.r.o.

Risk Disclosure

Risk Disclosure Statement

Trading any financial market involves risk. This Manual is neither a solicitation nor an offer to Buy/Sell any financial product. The contents of this Manual are for general informational purposes only.

Although every attempt has been made to ensure accuracy, the author does not give any expressed or implied warranty as to its accuracy. The author does not accept any liability for error or omission. All examples are provided for illustrative purposes only and should not be construed as investment advice.

No representation is being made that any account, or trader will, or is likely to achieve profits or losses similar to those discussed in this Manual. Past performance cannot be relied upon as being indicative of future performance.

The information provided in this Manual is not intended for distribution to, or use by any person or entity in any jurisdiction or country where such distribution or use would be contrary to law or regulation or which would subject the author to any registration requirement within such jurisdiction or country.

Hypothetical performance results have many inherent limitations, some of which are mentioned below. No representation is being made that any account will or is likely to achieve profits or losses similar to those shown. In fact, there are frequently sharp differences between hypothetical performance results and actual results subsequently achieved by any particular trading system.

One of the limitations of hypothetical performance results is that they are generally prepared with the benefit of hindsight. In addition, hypothetical trading does not involve financial risk and no hypothetical trading record can completely account for the impact of financial risk in actual trading.

For example the ability to withstand losses or to adhere to a particular trading program in spite of the trading losses is material points, which can also adversely affect trading results. There are numerous other factors related to the market in general or to the implementation of any specific trading program, which cannot be fully accounted for in the preparation of hypothetical performance results. All of which can adversely affect actual trading results.

U.S. Government Required Disclaimer - Commodity Futures Trading Commission Futures, Currency and Options trading has large potential rewards, but also large potential risk. You must be aware of the risks and be willing to accept them in order to invest in the futures and options markets. Don't trade with money you can't afford to lose. This is neither a solicitation nor an offer to Buy/Sell futures or options. No representation is being made that any account will or is likely to achieve profits or losses similar to those discussed on this web site. The past performance of any trading system or methodology is not necessarily indicative of future results.

CFTC RULE 4.41 - HYPOTHETICAL OR SIMULATED PERFORMANCE RESULTS HAVE CERTAIN LIMITATIONS. UNLIKE AN ACTUAL PERFORMANCE RECORD, SIMULATED RESULTS DO NOT REPRESENT ACTUAL TRADING. ALSO, SINCE THE TRADES HAVE NOT BEEN EXECUTED, THE RESULTS MAY HAVE UNDER-OR-OVER COMPENSATED FOR THE IMPACT, IF ANY, OF CERTAIN MARKET FACTORS, SUCH AS LACK OF LIQUIDITY. SIMULATED TRADING PROGRAMS IN GENERAL ARE ALSO SUBJECT TO THE FACT THAT THEY ARE DESIGNED WITH THE BENEFIT OF HINDSIGHT. NO REPRESENTATION IS BEING MADE THAT ANY ACCOUNT WILL OR IS LIKELY TO ACHIEVE PROFIT OR LOSSES SIMILAR TO THOSE SHOWN.

Warranty Disclaimer

You agree to use this program at your own risk.

StrategyQuant software might fail or not work properly. All software is subject to inadvertent programming errors and bugs embedded in the code comprising that software. Any of these errors and bugs can cause the software in which they are located to fail or not work properly. The StrategyQuant application and trading strategies generated by the program is subject to this risk. Despite testing, inadvertent errors and bugs may still cause a failure in trading strategy, resulting in a trading errors.

YOU HAVE TO BE AVARE THAT AUTOMATIC TRADING STRATEGIES MIGHT FAIL FOR WHATEVER REASON AND THEY COULD RESULT IN THE LOSS OF ALL OF THE MONEY YOU HAVE DEPOSITED IN THE BROKERAGE ACCOUNT YOU USE FOR LIVE TRADING BASED ON THE ALGORITHMS YOU GENERATED BY StrategyQuant. YOU SHOULD DISCUSS WITH AN INVESTMENT PROFESSIONAL THE RISKS OF TRADING IN GENERAL AND ALGORITHMIC TRADING IN PARTICULAR. YOU USE ANY ALGORITHM IN LIVE TRADING AT YOUR OWN RISK AND IT IS YOUR OBLIGATION TO THOROUGHLY AND APPROPRIATELY TEST ANY TRADING ALGORITHM BEFORE YOU PUT IT IN PRODUCTION AND TO CONTINUALLY MONITOR THE OPERATION OF ANY TRADING ALGORITHM IN PRODUCTION TO ENSURE IT IS RUNNING PROPERLY AND IN COMPLIANCE WITH ANY APPLICABLE RULES.

DISCLAIMER OF WARRANTY. THE SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE IS BORNE BY YOU.

LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, TORT, CONTRACT, OR OTHERWISE, SHALL AUTHOR OR ITS SUPPLIERS OR RESELLERS BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL OR PUNITIVE DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES. IN NO EVENT WILL AUTHOR BE LIABLE FOR ANY DAMAGES IN EXCESS OF AUTHOR'S LIST PRICE FOR A LICENSE TO THE SOFTWARE, EVEN IF AUTHOR SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. FURTHERMORE, SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION AND EXCLUSION MAY NOT APPLY TO YOU.

Software License Agreement

This legal document is an agreement between you, the end user ('User'), and StrategyQuant s.r.o. ('Author').

AGREEMENT - By installing StrategyQuant ('Software'), copying the Software and/or clicking on the 'I Agree' button during installation, you agree to all of the terms of this software license agreement ('Agreement'). If you do not agree with all of the terms of this agreement, click on the 'I Do Not Agree' button and/or do not install, copy or otherwise use the software.

INTELLECTUAL PROPERTY. The Software and any associated materials are protected by copyright law. The package is a proprietary product of the Author. The Author retains title to and ownership in the copyright of the software program and the associated materials. You acknowledge that the Author owns all rights, title and interest in and to the Software, including without limitation all Intellectual Property Rights. 'Intellectual Property Rights' means any and all rights existing from time to time under patent law, copyright law, trade secret law, trademark law, unfair competition law, and any and all other proprietary rights, and any and all applications, renewals, extensions and restorations thereof, now or hereafter in force and effect worldwide. You agree not to modify, adapt, translate, decompile, reverse engineer, disassemble or otherwise attempt to derive source code from the Software.

REGISTRATION. This program is neither freeware nor public domain. Use requires valid license. Contact us at <http://www.geneticbuilder.com/contactus/> for bulk licenses or discounted prices.

GRANT. Author hereby grants you a non-exclusive, non-transferable license to use the Software upon payment of the License Fee until the expiry date of the license (if any). Author makes no guarantee of the frequency, value, applicability or content of future updates or modifications to the Software. The Software will only be made available to you in electronic form for download.

The requirement to pay a license fee does not apply to evaluation copies for which Author does not charge a license fee. Evaluation licenses expire 14 calendar days from the date of this agreement, unless otherwise agreed to in writing by Author. On the date of expiry of the license, User agrees to either purchase the Software at the list price in force at that time or to destroy all copies of the Software in electronic or other form, including any copies on backup tapes or other media.

User's use of the Software shall be limited to use on a single hardware chassis, on a single central processing unit, as applicable, or use on such greater number of chassis or central processing units as User may have paid the required license fee.

You may not: permit other individuals to use the Software except under the terms listed above; translate, reverse engineer, decompile, decrypt, reverse engineer, disassemble (except to the extent applicable laws specifically prohibit such restriction), or create derivative works based on the Software; copy the Software (except for back-up purposes); rent, lease, transfer, assign, sub-license or otherwise transfer rights to the Software; or remove any proprietary notices or labels on the Software.

DISCLAIMER OF WARRANTY. THE SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SOFTWARE IS BORNE BY YOU.

LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, TORT, CONTRACT, OR OTHERWISE, SHALL AUTHOR OR ITS SUPPLIERS OR RESELLERS BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL OR PUNITIVE DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES. IN NO EVENT WILL AUTHOR BE LIABLE FOR ANY DAMAGES IN EXCESS OF AUTHOR'S LIST PRICE FOR A LICENSE TO THE SOFTWARE, EVEN IF AUTHOR SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY

ANY OTHER PARTY. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. FURTHERMORE, SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS LIMITATION AND EXCLUSION MAY NOT APPLY TO YOU.

TERMINATION. This license will terminate automatically if you fail to comply with the limitations described above. On termination, you must destroy all copies of the Software in electronic or other form, including any copies on backup tapes or other media. Upon termination of this License for any reason, you shall have no right to refund of the whole or part of any License Fee paid.

Table of Contents

1	Introduction.....	9
1.1	What is StrategyQuant?	9
1.2	What’s new in StrategyQuant X?	11
1.3	System Requirements.....	12
1.4	Installation.....	13
1.4.1	Post-Installation steps	13
1.5	Backtesting engines – Tradestation, NinjaTrader, MetaTrader4	15
2	How does StrategyQuant work?	16
2.1	Random generation of trading strategies	16
2.2	Genetic Evolution	17
2.3	Example strategy code	18
2.4	Supported building blocks.....	19
3	Quick start with the program	20
3.1	Main concepts	20
3.1.1	Program layout.....	20
3.1.2	Builder layout	22
3.1.3	Databanks and files	23
3.1.4	Different build modes.....	25
3.1.5	Strategy style	26
3.2	Workflow	28
3.2.1	Standard workflow	28
3.2.2	Cross checks – automated strategy robustness tests	31
4	Program screens.....	34
4.1	Builder	34
4.2	Settings.....	35
4.2.1	What to build.....	36
4.2.2	Parts to improve	43
4.2.3	Data	44
4.2.4	Trading options.....	47
4.2.5	Building blocks.....	49
4.2.6	Money management	53
4.2.7	Cross checks (robustness tests).....	55
4.2.8	Ranking options	55
4.2.9	Genetic options	59
4.3	Results	62
4.3.1	Databanks.....	62
4.3.2	Overview.....	63
4.3.3	List of trades	63
4.3.4	Equity chart	63
4.3.5	Trade analysis	64
4.3.6	Robustness tests - Monte Carlo	64
4.3.7	Strategy settings.....	65
4.3.8	Source code	66
5	Advanced functionality.....	69
5.1	Cross checks - robustness tests and analysis	69

5.1.1	Retest with higher precision.....	71
5.1.2	Monte Carlo trades manipulation	72
5.1.3	Retest on additional markets	74
5.1.4	Monte Carlo retest methods	74
5.1.5	Walk-Forward Optimization	75
5.1.6	Walk-Forward Matric	75
5.2	Optimization	77
5.2.1	Simple Optimization	77
5.2.2	Walk-Forward Optimization	85
5.2.3	Walk-Forward Matrix	86
5.3	Portfolios	87
5.4	Controlling conditions negation	88
6	How to...	90
6.1	Import history data from MetaTrader 4.....	90
6.2	Export strategy from StrategyQuant and test or trade it in MetaTrader.....	93
6.3	Test strategy in MetaTrader 4 with tick precision.....	98
6.3.1	Possible issue – Custom MT4 settings or missing symbol.....	100
6.3.2	Possible issue – Backtest doesn’t produce trades or there are errors in the Journal.	102
6.3.3	Possible issue – Modelling quality is n/a	102
7	Troubleshooting & Recommendations	104
7.1	When nothing gets build for a long time.....	104
7.1.1	Using Genetic evolution with too big population	104
7.1.2	Using Genetic evolution with too strict filters	105
7.1.3	Getting too many rejections based on Automatic filtering.....	106
8	Appendix.....	109
8.1	Ranking criteria.....	109
8.2	Entry and exit types.....	113
9	Final Words.....	115

1 Introduction

1.1 What is StrategyQuant?

StrategyQuant is a program that automatically generates new unique trading strategies for virtually any asset - forex, stocks, futures or ETFs.

Using StrategyQuant, you can find profitable trading strategies for virtually any market, any timeframe and any chart type. No programming or trading knowledge is required.

The resulting strategies can be saved with complete source code as a code that you can run on the supported trading platforms – like MetaTrader 4 / 5.

With StrategyQuant you can:

- **Generate an unlimited number of unique trading strategies for virtually any market or timeframe**
- Save your strategies as a **MetaTrader 4 / 5 Expert Advisor Tradestation or Ninja Trader strategy** with full source code – please note that for now we are working on support for MT4 /5 in SQ X, other trading platforms will follow shortly.
- Eliminate the manual labor previously required when developing a trading strategy
- **Find new trading strategies** that are not only unique, but also non-obvious
- **Reduce the time required to build a strategy** from weeks and months to minutes!
- Improve your existing strategies
- Optimize your strategies and find the best parameters
- Test your strategies for robustness and analyze their performance

Is StrategyQuant Right for You?

If you trade using automatic trading system (called also robots or Expert Advisors) or you plan to develop your own trading strategies, then **StrategyQuant can save you money and hundreds of hours of your time.**

Some traders prefer to purchase an existing trading robot; there is a multitude of offers especially for Forex.

While this can be an effective way, by purchasing someone's forex robot you are usually purchasing a black box - you don't know how it works, what are the exact rules and you are at mercy of its creator with potential adaptations to changed market conditions.

StrategyQuant allows you to **create your own trading strategies**, exportable to a plain Expert Advisor source code, so you have full control over your strategy.

What If You Are a Manual Trader?

You can still use for StrategyQuant to generate the trading ideas. You'll be surprised to find many profitable strategies based on relatively simple rules that you wouldn't think of.

Every strategy that is created by StrategyQuant can be exported to readable pseudo code with full description of the trading rules and can be traded also manually.

What to Expect

Please keep in mind that **StrategyQuant is a powerful tool, but it is not a magic box** that will start making you money with a click on a button. It has to be used in the right way to get the results.

Generating new strategies in StrategyQuant is only about 50% of the work.

The rest of the work has to go into evaluating the generated strategies to filter out the ones that are curve fitted or not robust enough.

It is up to you to evaluate your new strategies properly and know their strengths, weaknesses and limitations before you put them to live trading.

It can easily happen that from all the profitable strategies generated by StrategyQuant **only 1 out of 10 passes the evaluation** and we can consider using it for live trading.

But - the number of strategies we can generate is almost endless, so even 5-10% from infinity is a pretty big number :-)

There are few steps to evaluate the strategies quality and measure how good they will be in real live trading. Please read the **Evaluating Generated Strategies** section for more information.

Please note !!!

StrategyQuant X is still in development, and we are focusing on MetaTrader 4 / 5 in the first phase.

Support for Tradestation, MultiCharts, NinjaTrader and other platforms – although they might be mentioned in this manual – is still in development and it is not yet available in StrategyQuant X.

1.2 What's new in StrategyQuant X?

StrategyQuant X was rebuilt from the ground up to be able to accommodate new advanced features. Previous version 3 used a hard-coded strategy architecture that couldn't be altered, and its backtest engine was customized to support only this architecture.

Main new features in StrategyQuant X:

- **New UI** – based on HTML/JS, which is more intuitive and more flexible
- **It has full backtesting/trading engine** – SQ X now has a complete trading engine that can be used to test any strategy, it is not limited to a fixed strategy format like in SQ3.
- **Multi-timeframe or multi-symbol strategies** – SQ X now allows you to generate strategies that trade on multiple charts, be it the same symbol, just different timeframe, or a totally different symbols
- **Flexible strategy architecture** – SQ X allows you to choose from three strategy architectures, including one with fuzzy logic, and thanks to strategy templates you can easily generate your own type of strategy
- **Extendability** – you can extend SQ X to support your own indicators, building blocks or even things like computed statistical values
- **Custom projects allowing flexible tasks flow** – in addition to the three main parts of the program (Builder, Retester, Optimizer) you now have a possibility to build your own custom project with its own flow of tasks.
It allows to create automated processes that can run without user intervention and can continually build, test, verify and filter the generated strategies.
- **Support for grid and parallel computation on multiple computers** - this is something we are still developing, but it will be possible to use multiple computers to speed up the tests or generation of your trading strategies
- **Many other smaller enhancements** - new version also allows us to easily add new features in the future, and extend SQ X possibilities

1.3 System Requirements

StrategyQuant is not an EA, it is a normal program (EXE file) for Microsoft Windows and it will run on all standard computers **with Internet connection**.

Minimum system requirements:

- 1.2 GHz processor
- 2 GB RAM
- 500 MB hard disk space
- Windows XP, Vista, Windows 7, Windows 8 or Windows 10 64-bit operating system

Computing power

Strategy generation and backtesting requires a lot of processing power, so the faster computer you have, the more strategies will be generated and tested.

The most important component that affects the speed of testing is processor (CPU). StrategyQuant can use all processor cores, so the best results will be achieved by using multi-core processor, such as i5 or i7.

1.4 Installation

StrategyQuant comes with standard setup wizard, you just download and run the installation EXE file and then follow the steps in the installation wizard.

Important - Please DON'T install StrategyQuant to standard C:\Program Files directory !

It might not work correctly because Windows security settings don't allow the program to write to its data files.

Instead install it to any normal drive or directory on the disk, like C:\StrategyQuant or C:\Trading\StrategyQuant

1.4.1 Post-Installation steps

there are few things you should do after installation depending on trading platform you use.

Some of the technical indicators or functions used in StrategyQuant X are implemented specifically by us to implement them in a standard way to remove some edge cases that caused difference in backtesting.

All these additional indicators are a part of the installation package, you can find them in **{StrategyQuant}/custom_indicators** folder.

It is important to add all SQ X custom indicators to your trading platform !

MetaTrader4

Simply copy all the *.mq4 or *.mq5 files from this folder to your MetaTrader installation.

How to find the correct MetaTrader folder to copy the indicators to:

1. Open MetaTrader
2. Go to **File -> Open Data Folder** – this will open MT folder in explorer window
3. There go to **MQL4** or **MQL5**, then to **Indicators** subfolder
4. Copy all SQ custom indicators there
5. SQ indicators will be compiled and available in MetaTrader after you restart it

If you use more than one MetaTrader installations, you have to repeat this step for every MT4 installation you use or will use in the future.

Tradestation

Open your Tradestation program, go to **File -> Import/Export EasyLanguage** and choose **Import EasyLanguage File (ELD, ELS or ELA)**.

then find the file:

{StrategyQuant}/custom_indicators/Tradestation/SQ.eld and import it.

This will import all the additional functions StrategyQuant uses, so your new strategies will run in Tradestation.

1.5 Backtesting engines – Tradestation, NinjaTrader, MetaTrader4

StrategyQuant X can export trading strategies to multiple trading platforms:

- **MetaTrader 4 / 5** - forex, CFDs
- **NinjaTrader** - futures, stocks, forex – IN DEVELOPMENT
- **Tradestation** - futures, stocks, forex – IN DEVELOPMENT

There are some differences between these platforms in how each platform handles trades, opens and closes positions, manages the open trades, etc. **which can cause strategy to have very different results in one engine compared to the other. Always develop strategies in the engine you'll be trading it later!**

StrategyQuant allows you to switch the **backtesting engine** between MetaTrader / NinjaTrader / Tradestation so that internal testing engine knows how to treat the trades in a way that matches the selected platform.

Each trading platform and backtesting engine also allows you to configure some settings that are specific for a given platform.

Please note !!!

StrategyQuant X is still in development, and we are focusing on MetaTrader 4 / 5 in the first phase.

Support for Tradestation, MultiCharts, NinjaTrader and other platforms – although they might be mentioned in this manual – is still in development and it is not yet available in StrategyQuant X.

2 How does StrategyQuant work?

StrategyQuant is a program, it doesn't have the brain or experience of a trader, and it doesn't know how to create a profitable strategy.

What it does is that it randomly combines available building blocks (indicators, prices, etc.) to create new trading rules. The resulting strategy is then tested on a history data to see if it is profitable.

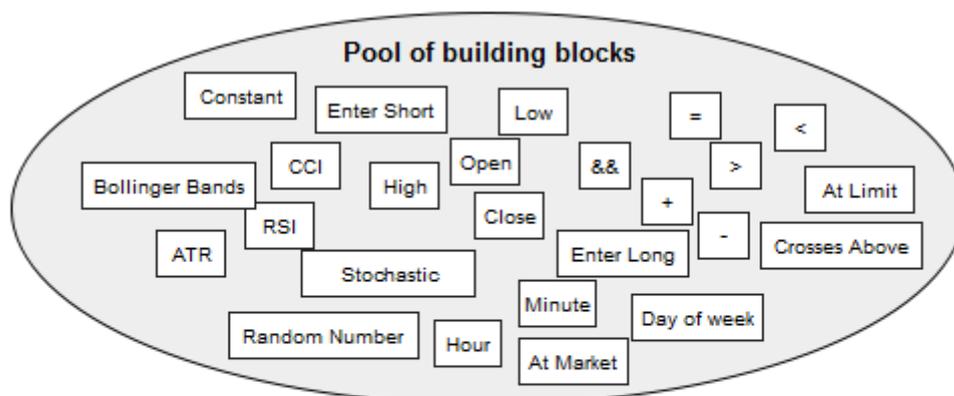
Random generation is the foundation of StrategyQuant. Strategies generated this way can be further improved (evolved) using **Genetic evolution**.

2.1 Random generation of trading strategies

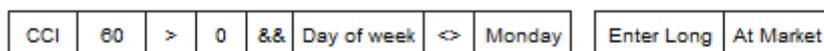
A trading strategy in the initial population is constructed using a combination of price patterns, technical indicators, order types, and other parts to form the entry and exit rules.

StrategyQuant can use all standard technical indicators and oscillators (like CCI, RSI, Stochastic, etc.), time values (like time of day, day of week) and price patterns. These building blocks are then combined using logical and equality operators (and, or, >, <, etc.) to form an entry or exit rule. In addition, it supports different entry and exit order types (market order, limit order, fixed profit target, exit after X bars, etc.).

With all the possible combinations of rules and orders, StrategyQuant is capable of generating literally trillions of different possible trading strategies.



Example of randomly generated entry rule:



if (CCI(60) > 0 && Day of week <> Monday) then Enter Long At Market

The building process itself is completely random - builder randomly picks different building blocks from the available pool and combines them to create entry rule, order type and exit rule.

There are some validity constraints that ensure that, for example price is not compared to time value, etc.

The result is a completely new random trading strategy. Of course, not every randomly created

strategy is profitable, but StrategyQuant can produce and test thousands of new strategies per hour, and there are many profitable ones in this amount.

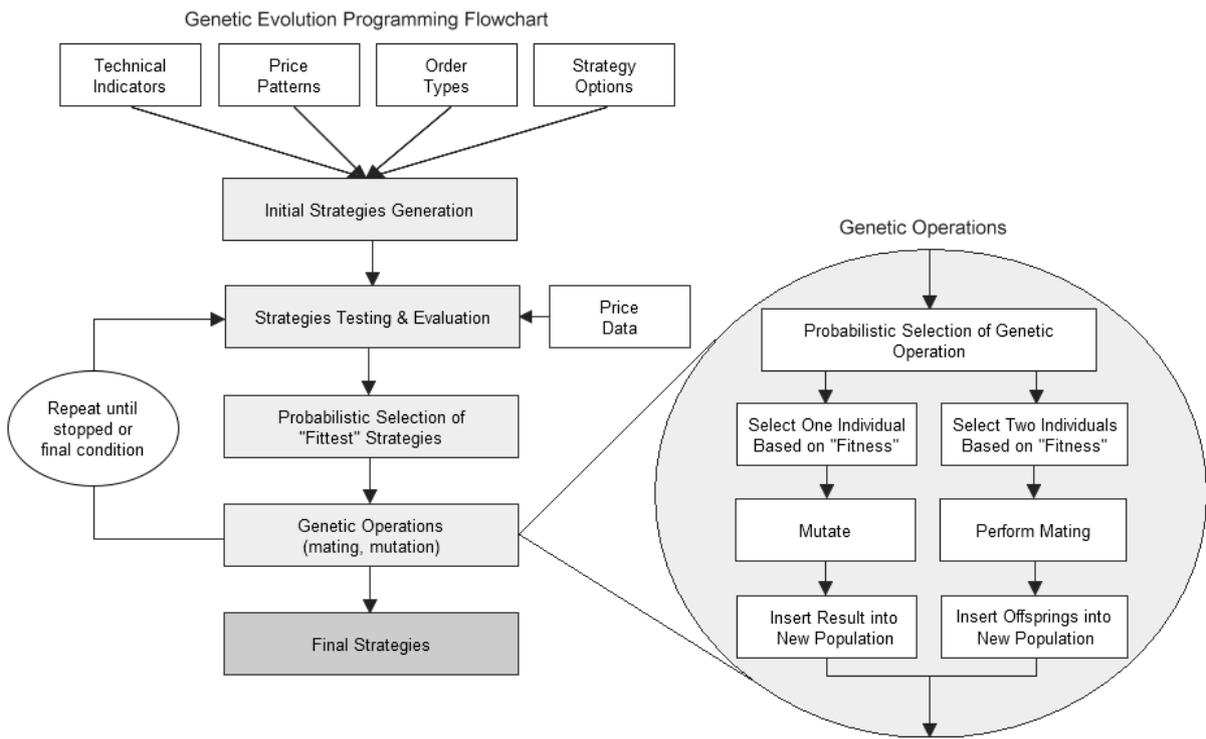
2.2 Genetic Evolution

Genetic Evolution takes the process of finding suitable trading strategies even further. In this mode StrategyQuant first creates a number of random strategies, which are used as the initial population in the evolution.

This initial generation of strategies is then "evolved" over successive generations using genetic programming technology.

This process imitates the evolution - the algorithm chooses the fittest strategies (using selected performance criteria) in every generation, and the group of fittest candidates is then used to produce new generation of trading strategies.

As in evolution, this should result in better and better candidates, in our case in strategies that are more profitable, more stable, or generally better in the selected performance criteria.



2.3 Example strategy code

Below is an example pseudo code of a strategy generated by StrategyQuant. You can see that strategy consists of entry orders, exit orders and trade management commands – such as trailing stop movements, etc.

Every strategy generated by the program can be viewed in this pseudo code or exported in the form of MetaTrader Expert Advisor (EA), NinjaTrader NinjaScript C# strategy or EasyLanguage for Tradestation/Multicharts.

```

=====
== Entry conditions
=====
LongEntryCondition = (Stoch(40, 1, 3) < 50)
ShortEntryCondition = (Stoch(40, 1, 3) > 50)

=====
== Entry orders
=====
-- Long entry
if LongEntryCondition is true {
    if No position is open then Buy at Ichimoku(6, 18, 38, Kijun-sen) + (0.4 * ATR(86)) Limit;
    Stop/Limit order expires after 34 bars.

    Stop Loss = 190 pips;
    Profit Target = (0.74 * ATR(87)) pips;

    // Move SL to BE (on close)
    Move Stop Loss to Entry price when in profit at least (77 * ATR(12)) pips;

    // Profit trailing (on close)
    Profit Trailing by 222 pips;

    // Stop trailing (on close)
    Move Stop to (Close(1) + (0.5) * BBWidthRatio(20, 2.0)) on bar close;
}

-- Short entry
if ShortEntryCondition is true {
    if No position is open then Sell at Ichimoku(6, 18, 38, Kijun-sen) + (-0.4 * ATR(86)) Limit;
    Stop/Limit order expires after 34 bars.

    Stop Loss = 190 pips;
    Profit Target = (0.74 * ATR(87)) pips;
}
=====

```

== Exit orders

```
=====
-- Long exit
if MarketPosition is Long {
  if (Bars Since Entry >= 33) {
    Close position at market;
  }
}
```

2.4 Supported building blocks

StrategyQuant supports over 250 building blocks, including all the standard technical indicators like CCI, RSI, Stochastic, Momentum, etc.

It also supports all standard order types – Market, Stop, Limit and advanced exit methods like Trailing Stop or Move Stop Loss to Break Even.

The best thing about new SQ X is that it allows you to create your own building block, or develop your own indicator that can be used to extend the program.

We will be continually adding new technical indicators and other features to the program

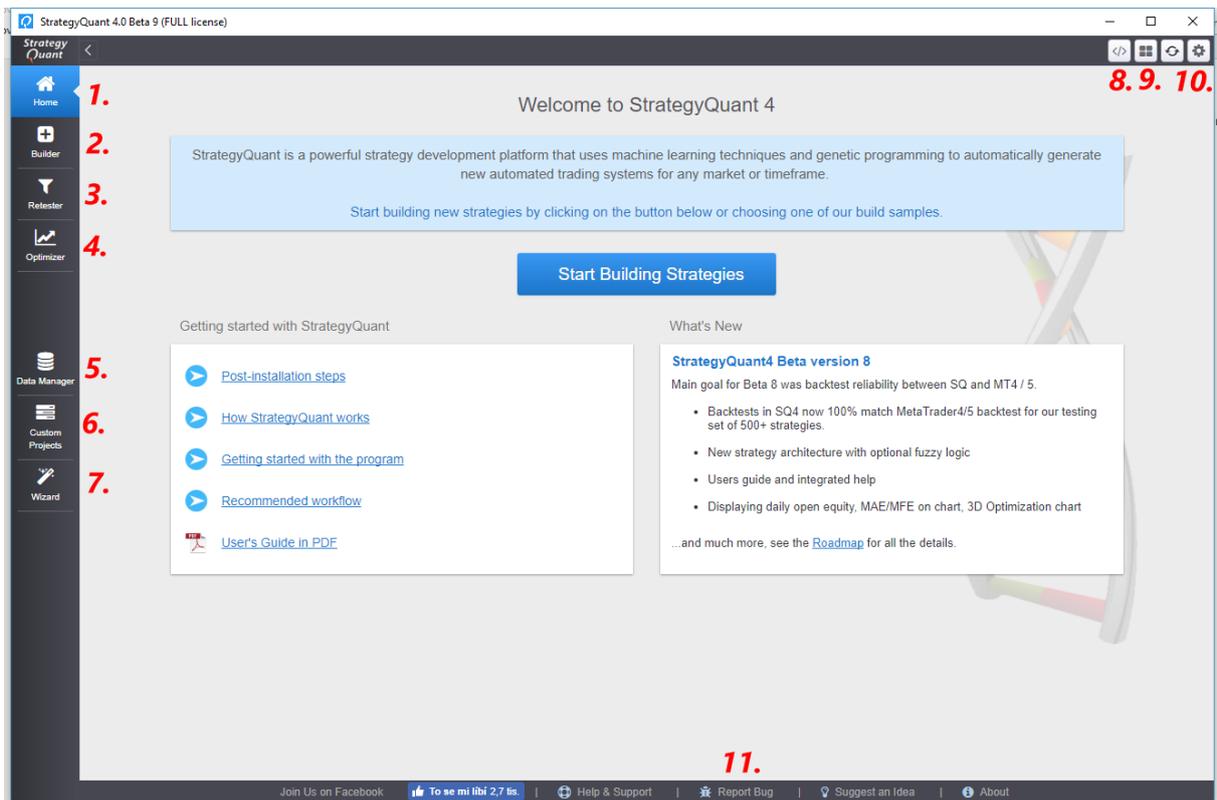
If you have your favorite indicator you'd like to see in StrategyQuant, just let us know.

3 Quick start with the program

3.1 Main concepts

3.1.1 Program layout

When you'll first start StrategyQuant you'll see the main screen as in the picture below.



The program functionality is divided into tabs on the sides.

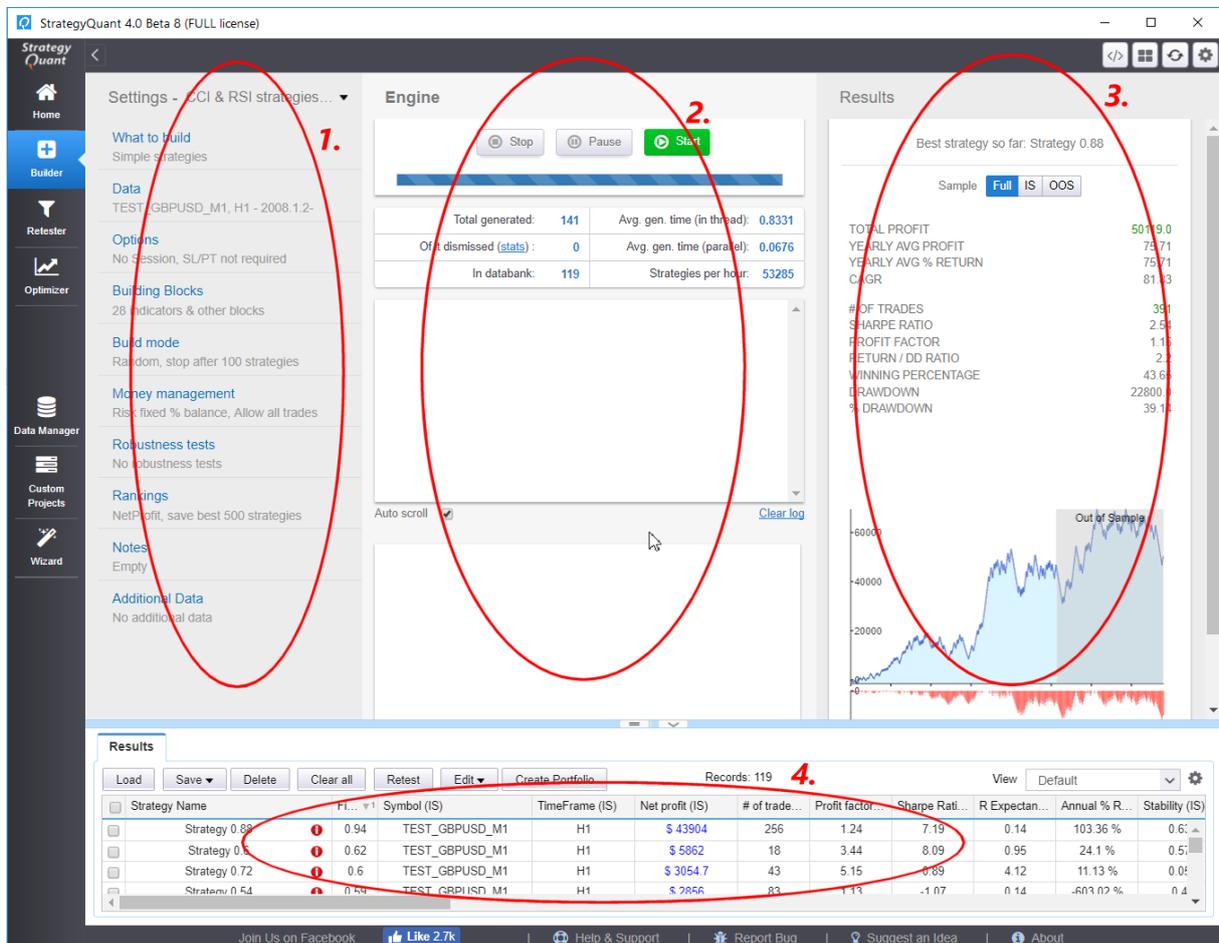
These tabs are:

1. **Home** - starting screen of the program, it contains the sample settings, news and helpful links
2. **Builder** – tool to build new strategies or improve existing ones. The core of the program.
3. **Retester** – tool to retest existing strategies on a different data or with different settings
4. **Optimizer** – tool to optimize parameters of existing strategies or to run Walk-Forward simulations
5. **Data manager** – tool to manage (download, import, export) data that are used in backtesting
6. **Custom projects** – tool to specify your custom flow of tasks

7. **Wizard** – editor of trading strategies. Here you can create a trading strategy from scratch by defining its exact rules
8. **Code Editor** – editor that allows you to build your own indicators or building blocks
9. **Grid Control** – overview of tasks running on computing grid
10. **Global config** – global program configuration options
11. **Report Bug** – report bug or suggest an idea using this link

3.1.2 Builder layout

When you click for example on Builder you'll see that the screen is divided into three main columns or sections:



- **Settings** – this is the brief description of actual configuration of Builder, you can click on each of these tabs to view the full configuration
- **Engine** – this is where you can start or stop the task, see what's going on in log and progress
- **Results** – up to three mini panels with results of the top three strategies currently existing in databank. These mini panels are only a preview of most important strategy statistics. You can get detailed results by clicking on this panel or double-clicking on a strategy in databank.
- **Databank** – a place where all strategies are stored

3.1.3 Databanks and files

Strategies in StrategyQuant are saved in their own proprietary file format (with .SQ X extension) that can be opened only by StrategyQuant.

If you find potentially good strategies you should always save them so that you can work with them later.

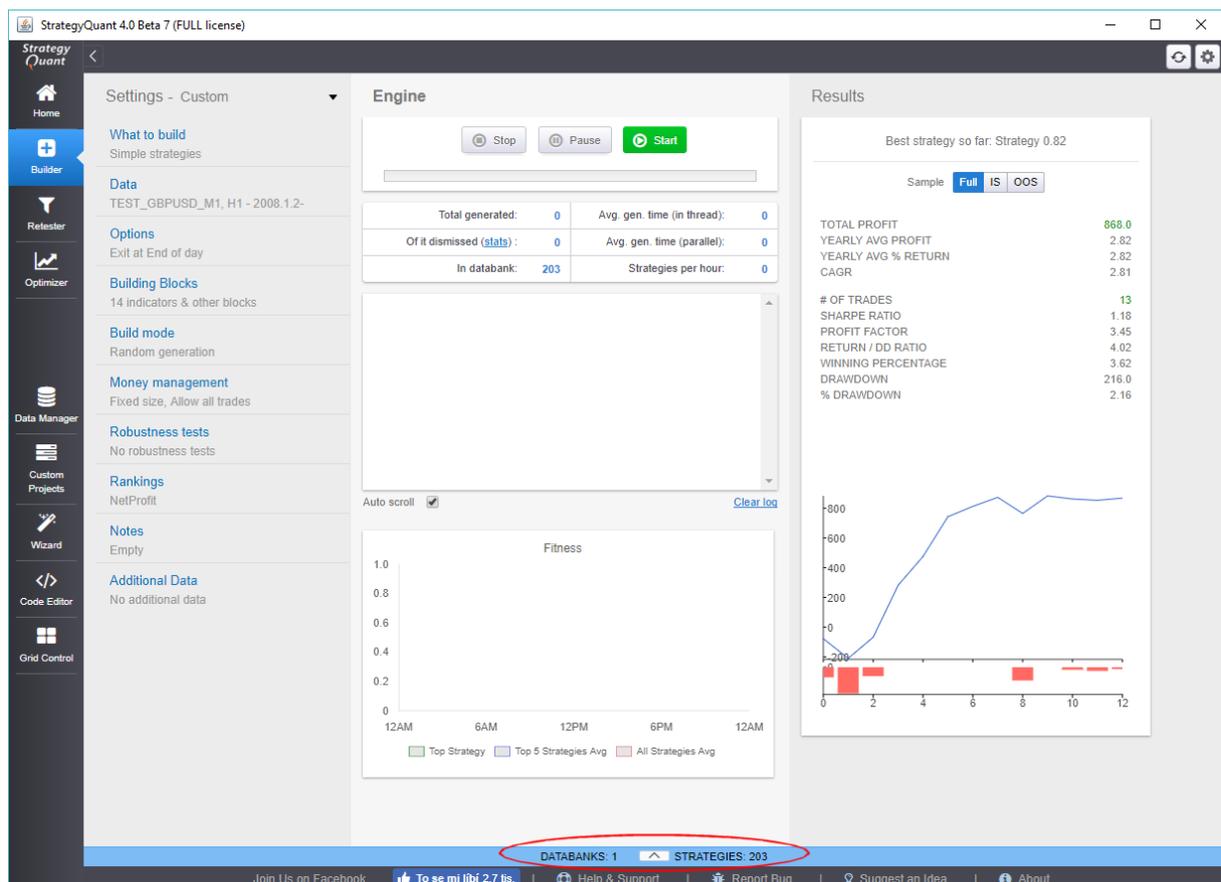
Running strategies in MetaTrader

MetaTrader cannot read the strategy .SQ X files. If you want to test or run your new strategies in MetaTrader you have to export the strategy to MQL source code.

Please check the section [How To ... Export strategy from StrategyQuant and test it in MetaTrader](#).

Please note that exported (*.MQ4) files are not readable by StrategyQuant, so make sure you always save your strategy also as a normal strategy file (*.SQ X).

Databank is the most important concept you should understand when using StrategyQuant. Doesn't matter which mode you use, strategies are always stored in databanks. Databanks are on the bottom side of the screen, it can be minimized or opened.



Minimized databank

The screenshot shows the StrategyQuant 4.0 Beta 7 (FULL license) interface. The 'Engine' section is active, showing 'Stop', 'Pause', and 'Start' buttons. The 'Results' section displays a summary of the best strategy so far: Strategy 0.82. Below this, a table lists various strategies with their performance metrics. A red circle highlights the 'Results' table.

Strategy Name	Fin...	Symbol (IS)	TimeFrame (IS)	Net profit (IS)	# of trade...	Profit factor...	Sharpe Rat...	R Expectan...	Annual % ...	Stability (IS)	Syr
Strategy 0.82	0.42	TEST_GBPUSD_M1	H1	\$ 866	13	3.15	1.18	39.97	2.62 %	0.28	
Strategy 0.71	0.4	TEST_GBPUSD_M1	H1	\$ 738	31	1.55	0.95	3.04	2.5 %	0.03	
Strategy 0.190	0.39	TEST_GBPUSD_M1	H1	\$ 647	10	2.41	1.27	4.66	2.18 %	0.16	
Strategy 0.52	0.39	TEST_GBPUSD_M1	H1	\$ 598	22	1.76	1.32	15.8	2 %	0.23	
Strategy 0.196	0.38	TEST_GBPUSD_M1	H1	\$ 580	20	2.11	1.79	8.98	1.93 %	0.21	
Strategy 0.13	0.38	TEST_GBPUSD_M1	H1	\$ 569	14	2.02	0.62	5.32	1.94 %	0.01	
Strategy 0.48	0.36	TEST_GBPUSD_M1	H1	\$ 428	3	5	2.29	0	1.41 %	0.2	
Strategy 0.170	0.35	TEST_GBPUSD_M1	H1	\$ 371	3	5	2.1	0	1.22 %	0.03	
Strategy 0.189	0.35	TEST_GBPUSD_M1	H1	\$ 321	28	1.22	0.34	0.78	1.15 %	0	

Open databank

All the strategies that program produces or is working with are in one of the databanks. There you can sort the strategies by its properties, load and save them, and when you double click on a row, it will open the strategy test details in the detailed Results window.

3.1.4 Different build modes

There are two build modes to choose from:

Genetic Evolution

StrategyQuant first generates initial population of random candidates (using the Random Generation mode) and then uses genetic evolution process to evolve the population and produce better and better candidates with each generation.

The process ends when predefined number of generations is reached or when there's no further improvement.

Pros:

- in theory it should lead to strategies better than the initial random generation
- this means that the already good strategies in the first generation can be further improved
- search for profitable strategy in the trillions of possible combinations can be more effective with the power of genetic evolution

Cons:

- evolution can be slower
- sometimes the evolution can lead to the dead end, so the generation should be watched
- the group of generated strategies is limited by population size

Random Generation

In this mode StrategyQuant continually generates and tests new random strategies, one after another, until it is stopped.

The top candidates (based on predefined criteria) are stored into Databank so you can review them later.

Pros:

- faster and simpler than genetic evolution
- it can run until it is stopped, so if you let it run for a few days it can generate and evaluate millions of strategies

Cons:

- once the strategies are generated they are not further evolved – but you can always use them as an initial population for the next genetic evolution

3.1.5 Strategy style

StrategyQuant X allows you to choose from 3 different strategy “styles”. By style we mean how the strategy is constructed.

Every trading strategy consists of set of **IF – THEN** rules, managing **IF** something happens **THEN** do some action. There are however some differences in how exactly these rules are constructed.

The three types are:

3.1.5.1 SQ3 (old) style

In previous version of SQ the generated strategies looked like this:

LONG ENTRY RULE: IF Long Entry Conditions THEN Open Long order

SHORT ENTRY RULE: IF Short Entry Conditions THEN Open Short order

LONG EXIT RULE: IF Long Exit Conditions THEN Close Long order

SHORT EXIT RULE: IF Short Exit Conditions THEN Close Short order

It is a simple and logical format, but what if both Long and Short conditions are valid at the same time? Then you’d have to open both Long and Short order, or the Short order cancels the Long one.

Or what if both Long entry and Long exit conditions are valid at the same time? Then you wouldn’t know whether to entry of exit.

This leads to problems that are covered by the new SQ X architecture.

3.1.5.2 SQ X new style

In SQ X new architecture the first rule is a special Signal rule that checks all the trading conditions. The rest of the rules then check the produced trading signals and open or close trades. The strategy looks like this:

*SIGNAL RULE: LongEntrySignal = Long Entry Conditions
ShortEntrySignal = Short Entry Conditions
LongExitSignal = Long Exit Conditions
ShortExitSignal = Short Exit Conditions*

*LONG ENTRY RULE: IF LongEntrySignal = true and ShortEntrySignal = false
and LongExitSignal = false
THEN Open Long order*

*SHORT ENTRY RULE: IF ShortEntrySignal = true and LongEntrySignal = false
and ShortExitSignal = false
THEN Open Short order*

*LONG EXIT RULE: IF LongExitSignal = true and LongEntrySignal = false
THEN Close Long order*

*SHORT EXIT RULE: IF ShortExitSignal = true and ShortEntrySignal = false
THEN Close Short order*

This ensures that strategy trades only when there is clear signal to one of the direction, it does nothing when the signals are contradicting.

3.1.5.3 SQ X new style with Fuzzy logic

A modification of the new rule is to employ fuzzy logic, the only difference from previous type is in Signal rule. Normally the conditions for the signal are connected with AND and OR.

The typical signal in a standard trading strategy could be something like:

SIGNAL RULE:

*LongEntrySignal = ((CCI(14) > 0) and (RSI(20) > 50))
or
((MACD(10, 20, 30) > 0) and Hammer Candle Pattern))*

With fuzzy logic we are adding possibility to evaluate all the conditions, and let some of them be wrong, while still having valid signal.

SIGNAL FUZZY RULE – 70% have to be right:

LongEntrySignal:

*CCI(14) > 0
RSI(20) > 50
MACD(10, 20, 30) > 0
Hammer Candle Pattern*

Note that we don't use any AND or ORs here, all conditions are evaluated.

With fuzzy rules you define one more thing - how big % of all the conditions has to be right to have the whole signal still valid.

In our case we have four conditions, and 70% of them (which means 3 conditions out of 4) have to be true.

So if **any three of those four conditions are true**, LongEntrySignal will be true.

Using fuzzy rules makes sense only if you'll let program generate more than just 2-3 sub-conditions in every signal, it doesn't make much sense if you'll have only 2 or 3 conditions in a signal.

But imagine if your LongEntrySignal consists of 10 conditions. Fuzzy logic can be seen as "voting", and majority of the conditions in the right direction will determine the outcome of the signal.

This opens new possibilities for strategies creation.

3.2 Workflow

3.2.1 Standard workflow

Please check the Articles section on our website for the most up-to-date tips on how to use the program:

<https://docs.strategyquant.com/>

The general flow of work when generating new strategies can be described as a series of the following steps:

1. Import or download data for backtesting
2. Configure Builder options
3. Run build
4. Evaluate generated strategies
5. Retest or optimize, perform more checks

1. Import or download data for backtesting

You can use the history data that come with the program, import your own data in various formats or download real tick data from Dukascopy broker.

2. Configure Builder options

Go through all the settings and configure strategy type, indicators, and order types to be used for trading rules. Optionally use time constraints to limit trading to a certain time range.

Turn on cross checks (strategy robustness tests) you want to use during strategy generation. Selected cross checks will be applied on every generated strategy and can automatically filter out “bad” strategies. You can learn more about cross checks in the next chapter.

Configure also ranking options – they allow you to select Strategy Selection Criteria - which is how the best strategies are determined.

You should also set up Custom Conditions to filter only strategies that pass certain criteria.

It makes sense to dismiss all the strategies that have too little profit or trades, or too small Profit Factor, Return/DD ratio or System Quality Number.

Configuring data – to make the generation phase as fast as possible, you can use Selected timeframe as the test precision option. This allows the program to run quickly and go through as many new strategies as possible. When you’ll find potential good candidates, you can test it with more exact precisions later.

3. Run Build

Start the building process. Depending on your settings you can let it run several minutes, several

hours or even several days. The more time it will run the more potential strategies it will test. The best of them will be always stored into the databank.

4. Evaluate generated strategies

Go through the generated strategies and evaluate them. You can either evaluate them visually by checking their equity chart, or by sorting them by their parameters in the Databank. Choose the best ones to pass to the next step and save them as StrategyQuant file (.SQ X) so that you can work with them later.

5. Retest or optimize, perform more checks

The goal of strategy evaluation is to find strategies that are robust and have real edge on the market.

It is not difficult to generate strategies that will have great equity curve, because it will be overfitted to the given history data.

Robust strategy should work in different conditions and doesn't break up when there is a small change in parameters or in price data or they miss few trades.

Generating good strategy (up to the point 4.) is only half of the job. The other half is to make sure the newfound strategy is “real”, and not overfitted.

To do that, you can use [Robustness Tests](#) as a part of the building process, but also later. You should retest your strategies on different markets and with different settings, and they should not fail. Only then you can get better assurance that the strategy is robust and will not fail in live testing.

This final step will consist of multiple stages, you can retest your strategies with different settings, on different markets and/or timeframes, or with different spread and slippage, on running Walk-Forward optimization or Walk-Forward matrix, etc.

Optional additional steps are:

6. Improve the strategy

You can try to improve the strategy in Improver. You can try to apply different combinations of exit rules or additional conditions to entry rules in search for better performance.

After the improvement you should again run the new strategy variation through robustness tests to make sure it didn't lose its robustness.

6. Optimize the strategy

You can run [simple optimization](#) to find better combination of input parameters of your strategy. You can also run [Walk-Forward optimization](#) to find out if the strategy would benefit from periodical reoptimization.

As the last step, you can run [Walk-Forward Matrix](#) analysis to determine the best reoptimization period.

There are articles on our website that describe more detailed flow of work:

<http://www.strategyquant.com/articles/getting%20started%20with%20StrategyQuant>

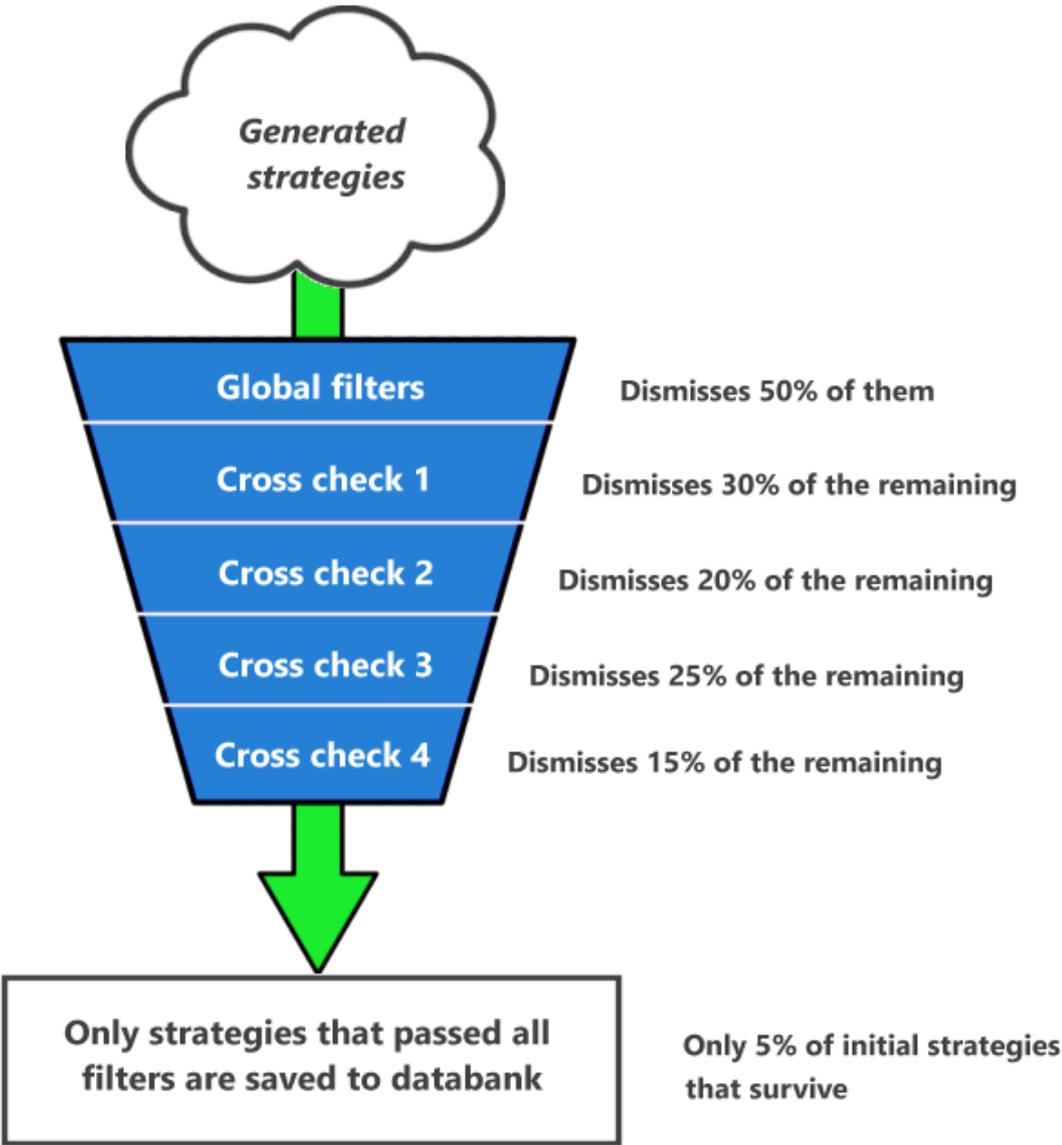
<http://www.strategyquant.com/articles/strategy%20building%20process>

3.2.2 Cross checks – automated strategy robustness tests

Cross checks are optional additional methods that can be applied to every strategy after it is generated and passes the first filters.

They can check strategy robustness from more points of view – by trading it on additional markets, or by using Monte Carlo methods to simulate 100s different equity curves, or even using Walk-Forward optimization or Matrix.

The important thing is that you can use cross check filters to dismiss the strategy if it doesn't pass the cross-check test. This allows you to create funnels, where strategy is scrutinized by increasingly advanced (and more time demanding) methods, and strategy that fails is automatically dropped.



An example of filtering funnel using cross checks during build

It is up to you how many cross checks you'll employ and how you'll configure their filters.

Cross checks are divided into three groups – Basic, Standard, Extensive – depending on how time consuming they are. They are also applied from the simple ones to the more complicated ones. So, if strategy doesn't pass Cross check 1, it is dismissed and not tested by Cross check 2.

Note that running cross check on a strategy can take significant time!

Some cross-check methods make complicated simulations and hundreds or even thousands of backtests of the strategy with different parameters and take thousands time more time than the initial strategy generation and initial backtest.

So strategy without any cross-check can be generated (for example) in 0.2 second, but with some cross checks applied it could easily take 10 - 200 seconds per one strategy!

Cross checks can be used also in Retester (without filtering), so you don't need to use all cross-checks in build mode.

Possible optimal usage of cross-checks described

The optimal cross-check application could be to use:

- Cross check **Retest with higher precision**
- Cross check **Monte Carlo trades manipulation**
- Cross check **Retest on additional markets**
- Optionally cross check **Monte Carlo retest methods**

StrategyQuant will then perform the following steps for every generated strategy:

1. Strategy is randomly generated and tested with fastest "Selected timeframe" precision – this is dependent on your setting, but it is the default
2. Strategy is automatically filtered and thrown away (dismissed) if it doesn't pass your global filters, for example, if it doesn't have enough trades or if Net Profit < \$1000
3. Cross check **Retest with higher precision** will retest this strategy with minute or even real tick precision – to make sure the strategy was reliably backtested using the basic precision. Note that only strategies that pass point 2. will get here. If strategy doesn't pass this first cross-check, it is dismissed.
4. Cross check **Monte Carlo trades manipulation** will run number of simulations of different equity curves by manipulating the existing trades – to ensure that the original equity curve wasn't achieved

just a luck. You'll filter out strategies that don't pass this Monte Carlo test

5. Cross check **Retest on additional markets** will test the strategy on additional markets or timeframes. If it isn't profitable on other markets it is filtered out.

6. Optional cross check **Monte Carlo retest methods** will run number of simulations where each simulation is a new backtest of the strategy using small variations in strategy indicator parameters, trading options such as spread, slippage, or in history data.

Note that every such simulation is an independent backtest, so if it took 0.2 seconds to backtest the strategy in step 1., it will take $100 \times 0.2 \text{ s.} = 20$ seconds to finish this cross check for this one strategy if you'll use 100 Monte Carlo simulations.

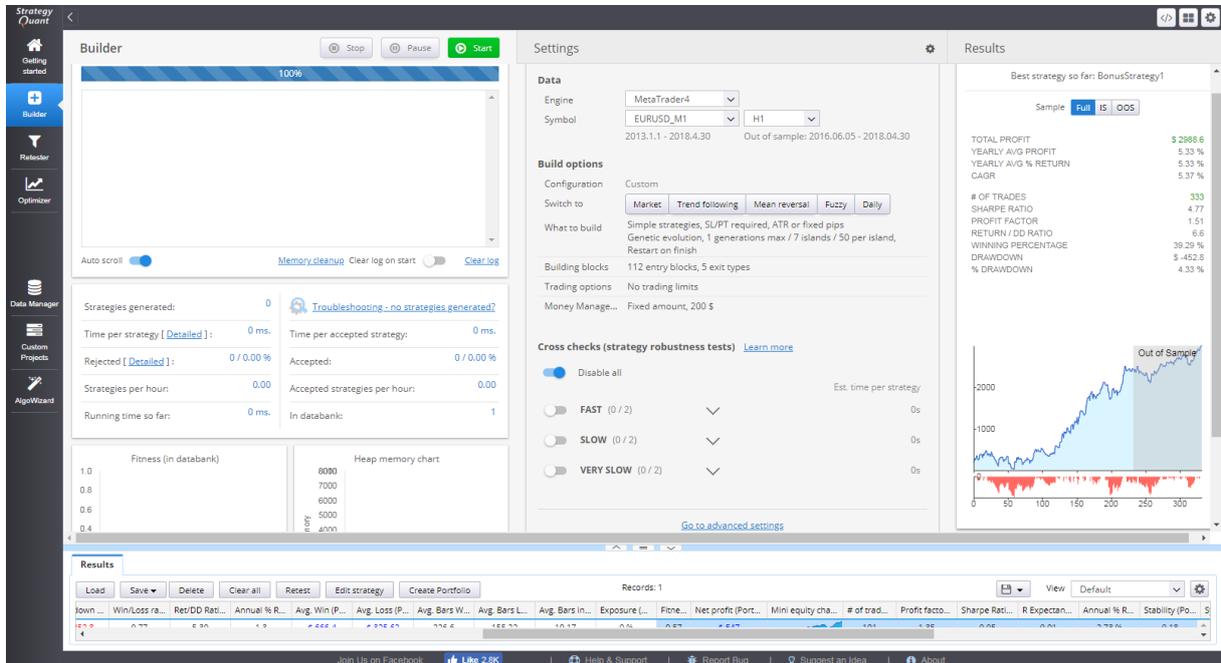
If strategy passes also this cross-checks it is saved into databank and you can have quite good confidence that it is robust enough.

You can find more detailed description of cross checks in the section Cross checks - robustness tests and analysis.

4 Program screens

4.1 Builder

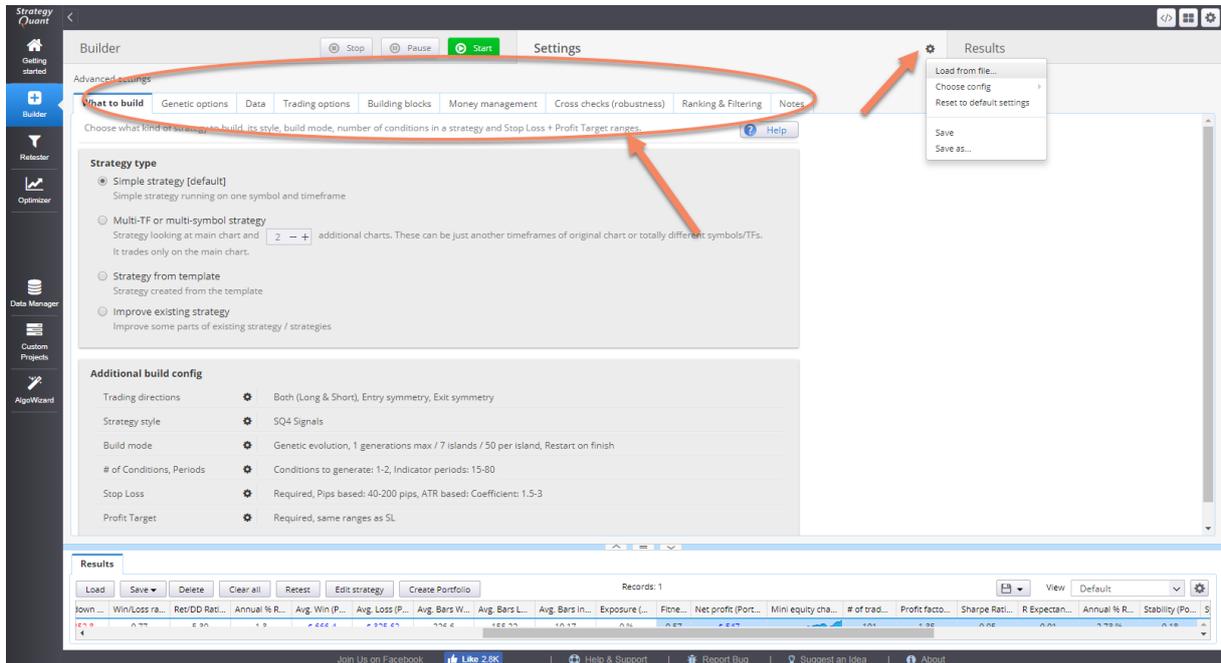
Builder is the “core” of the program, here you can build new trading strategies.



Before starting the build or test you should have the data and settings configured. The results (generated top strategies) will be continually stored in the **databank** on the bottom.

4.2 Settings

Settings for every part of the program are displayed in tabs on the top row.

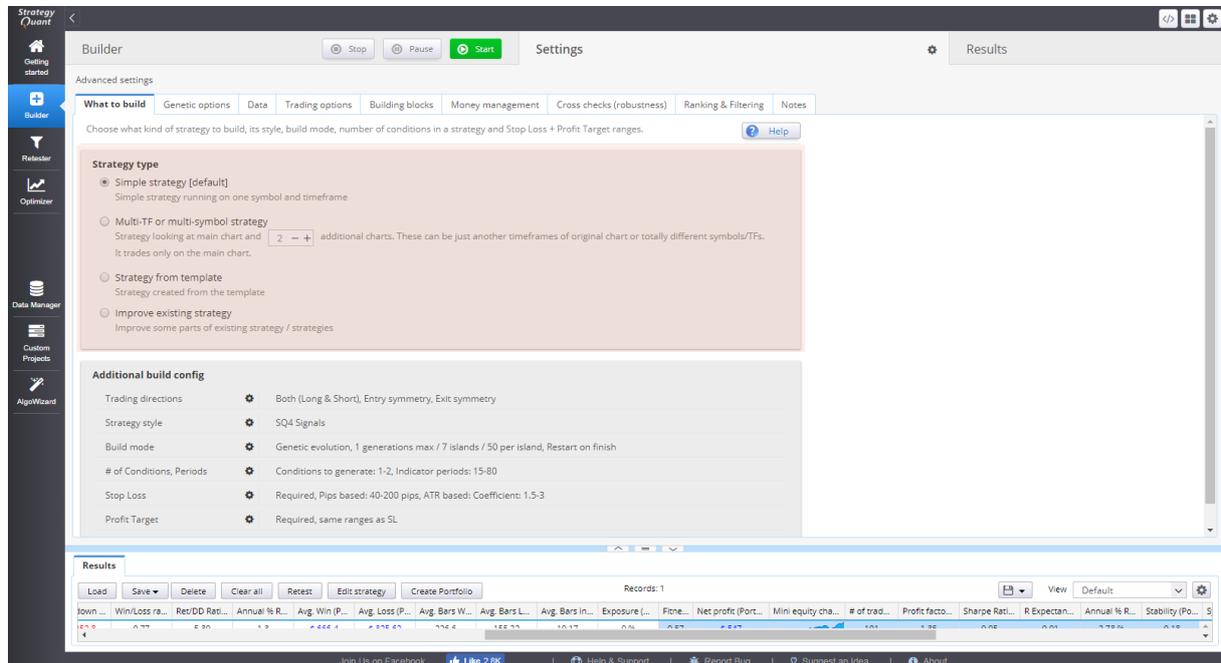


You can see more detailed settings by clicking to one of the tabs, it will open panel with settings details.

You can also save your current settings to a file and load it from the file later using the action button on the top right corner of settings column.

4.2.1 What to build

Here you select what exactly you want to generate. You can generate a new strategy for single chart, for multi-symbol or multi-timeframe, or improve a part of the existing strategy



Strategy type options:

- **Simple strategy** – “standard” simple strategy that trades on one symbol and timeframe
- **Multi-TF or multi-symbol strategy** – strategy that can use multiple additional charts in addition to the main one. For example, it will be trading on EURUSD/H1, but it could be looking also at data for EURUSD/H4 and GBPUSD/H1.
You simply must define how many of these additional charts the strategy will use, you’ll then later define which ones they will be exactly in Data settings
- **Strategy from template** – allows you to generate strategy using your own strategy template. You can create strategy template in Wizard, and then choose it here. This allows you to generate strategies with architecture that is different than standard SQ X strategies
- **Improve existing strategy** – you must choose strategy that you want to improve, and then in another settings **Parts to improve** you’ll choose what exactly should be improve in this strategy – it could be only Long entry rule, or only Short Exit rule, or type of order placed (Market / Stop / Limit).

Trading direction

You can choose to generate strategies that trade only to one direction (Long or Short) or to both directions (which is standard).

You can also select that you want the entry or exit rules to be symmetrical. If they are symmetrical, then the rules for both directions are the same, only reversed.

An example of symmetrical rules:

Go Long if CCI > 0

Go Short if CCI < 0

As an alternative, you can choose to use non-symmetrical rules, in this case the rules for Long and Short sides will be generated independently.

An example of non-symmetrical rules:

Go Long if CCI > 0

Go Short if RSI < 0 and Momentum < 100

This setting can be used for both entry and exit rules, for example you can have symmetrical entry rules, but non-symmetrical exit rules, so the strategy will effectively use (for example) different stop loss and profit target for Long and Short orders.

Strategy style configuration and build mode

There are important configuration options that you can set in addition to the basic strategy types. Clicking on this link will open a new popup dialog. You can choose from three different architecture types, they are described in detail in their own section in this guide.

Strategy style settings
✕

Strategy style

SQ4 Signals Style

SQ4 Signals Style with Fuzzy Logic

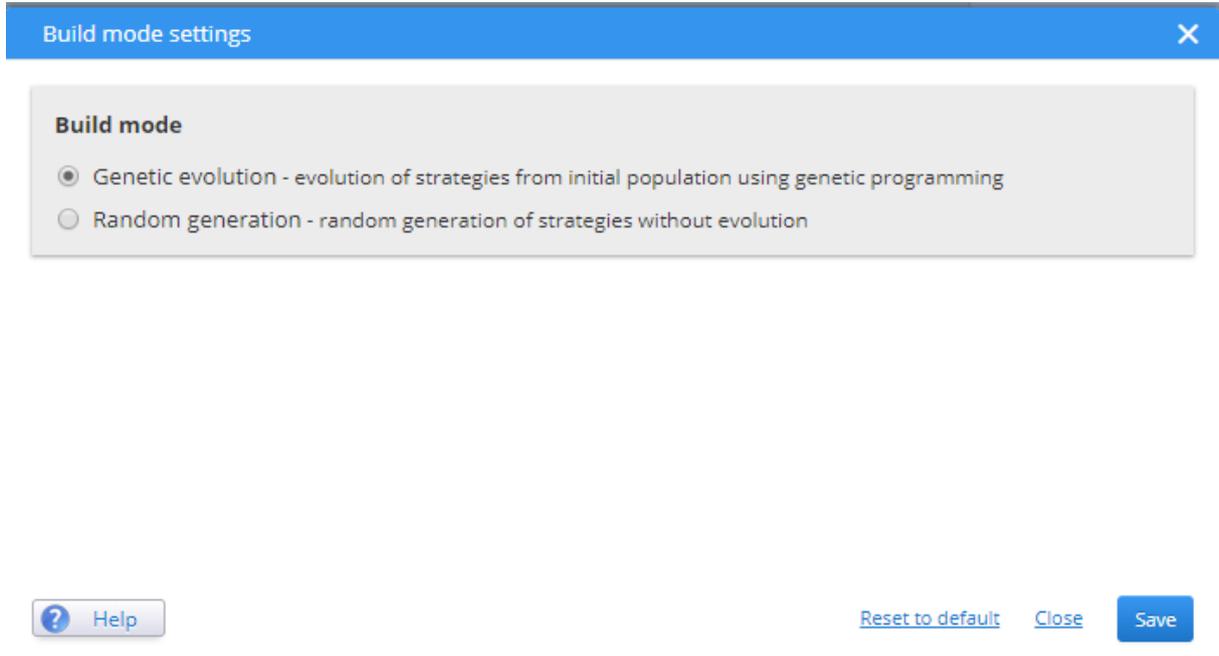
Minimum % of true conditions Min: -- + Max: -- +

Old SQ3 Style

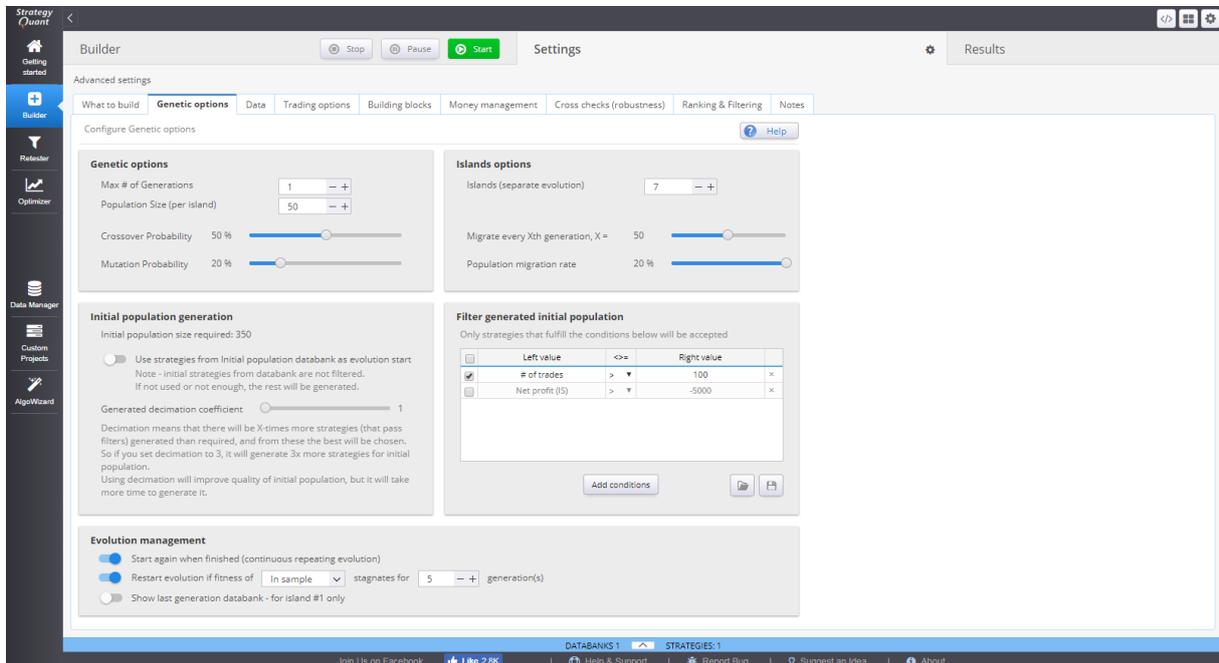
[? Help](#)
[Reset to default](#) [Close](#)
Save

Build mode

You can select whether SQ will generate strategies using **Genetic evolution** or **Random generation**.



If the genetic evolution option is used a new navigation tab "**Genetic options**" will show up in the builder. All the options available are discussed later in this guide.



Conditions to generate

This determines min and max number of conditions that should be generated for one signal.

of Conditions, Periods settings

Main chart

Conditions to generate	Min: 1 -- +	Max: 2 -- +
Global Shift (lookback period)	Min: 1 -- +	Max: 3 -- +
Global Indicators period	Min: 15 -- +	Max: 80 -- +

Help [Reset to default](#) [Close](#) [Save](#)

For example, if you'll allow only one condition your signal will look like:

$EntrySignal = CCI > 0$

If you'll use three conditions, it could look like:

$EntrySignal = CCI > 0 \text{ and } RSI > 50 \text{ or } Momentum < 0$

Note that there are three different conditions (CCI, RSI, Momentum comparison), connected with and/or.

Setting higher range for number of conditions is especially important for Fuzzy rules strategy, because fuzzy logic will be effective when there are at least 3 or four conditions to evaluate. So if you are using Fuzzy Logic architecture, make sure you set Conditions to generate Minimum at least to 3 or more.

Shift (lookback period) is the number of bars in the past the condition could look into. Shift=0 means that the condition is evaluated on the current bar, Shift=1 means the condition is evaluated on the previous bar, Shift=2 means it is evaluated on bar before previous one and so on. So here you can define ranges that will be used when generating strategy conditions.

In general, it isn't good to allow strategy to look too far into the past. For example, value of CI(14) 10 bars back doesn't bear too much significance for the current market status. It is recommended to keep this range small, between 0-5.

Using Minimum Shift = 0 ?

When you set minimum Shift = 0 you allow to create conditions that check for the indicator value on the latest current bar. Conditions are usually evaluated on bar open, but most of the indicators use

bar close to compute their value. So the value of indicator on bar open (For example CCI(14) could be very different from its value at the end of the bar.

For this reason we think it makes more sense to set Minimum Shift to 1, this way on bar open (when conditions are evaluated) you'll get indicator values from the previous bar that just finished, and the indicator value was computed and final.

Indicators period is min and max range of period value, meaning how big period should be used in indicators generated in StrategyQuant.

Indicators period must be bigger than 1, and ideally smaller than 100, or even smaller than 50. Again, it is recommended to not use too big periods.

If you use more than one chart, you can configure all these settings for each chart separately.

Note that all these settings are ranges from Min or Max. The exact number of conditions, shift and period will be determined randomly when every strategy or condition is generated.

The second part is configuration for every chart that the strategy can access. For simple strategy there will be only one chart, but if you'll build multi-TF or multi-symbol strategy, you can configure the generation for every chart separately.

of Conditions, Periods settings
✕

Main chart

Conditions to generate	Min: <input type="text" value="1"/> -- +	Max: <input type="text" value="2"/> -- +
Global Shift (lookback period)	Min: <input type="text" value="1"/> -- +	Max: <input type="text" value="3"/> -- +
Global Indicators period	Min: <input type="text" value="15"/> -- +	Max: <input type="text" value="80"/> -- +

Additional chart 1

Conditions to generate	Min: <input type="text" value="0"/> -- +	Max: <input type="text" value="2"/> -- +
Global Shift (lookback period)	Min: <input type="text" value="1"/> -- +	Max: <input type="text" value="3"/> -- +
Global Indicators period	Min: <input type="text" value="15"/> -- +	Max: <input type="text" value="100"/> -- +

Additional chart 2

Conditions to generate	Min: <input type="text" value="0"/> -- +	Max: <input type="text" value="2"/> -- +
Global Shift (lookback period)	Min: <input type="text" value="1"/> -- +	Max: <input type="text" value="3"/> -- +
Global Indicators period	Min: <input type="text" value="15"/> -- +	Max: <input type="text" value="100"/> -- +

Use different settings for every chart

? Help

[Reset to default](#)
[Close](#)
[Save](#)

Stop Loss & Profit Target Options

These settings allow you to specify whether Stop Loss and Profit Target should be mandatory in the strategy, and what is the minimum and maximum of the SL/PT values in pips.

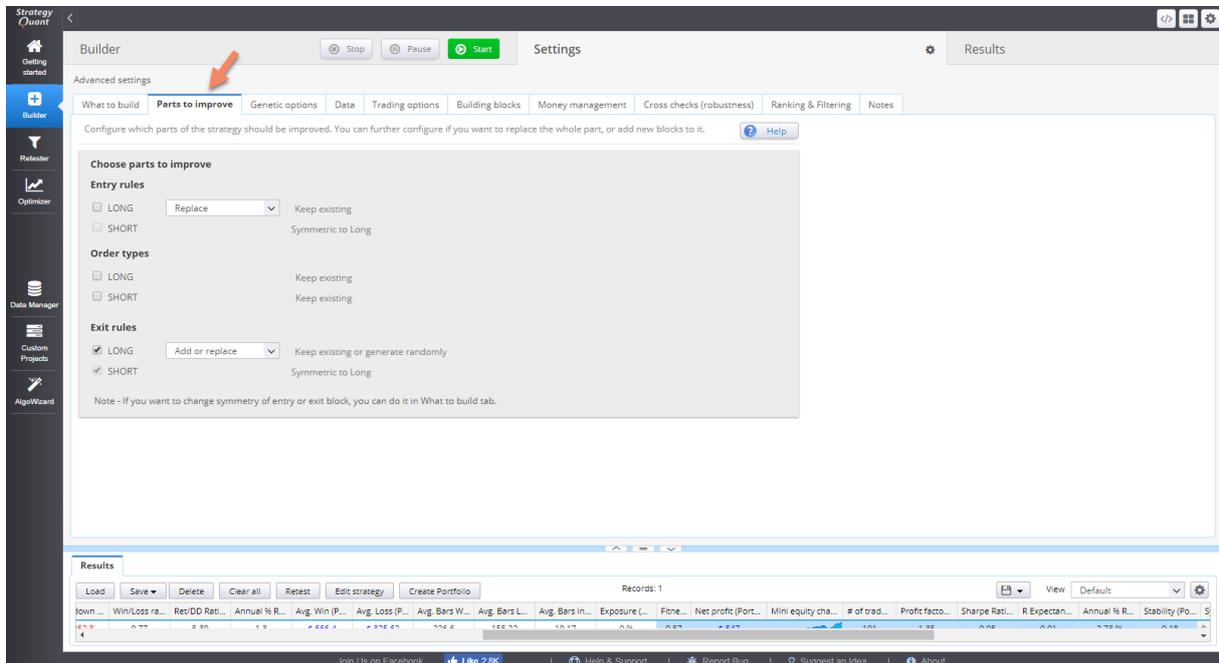
You can also define the desired Risk Reward ratio.

Having defined SL/PT in the strategy is the simplest and many times the most effective approach.

If you unselect the mandatory SL/PT then the randomly generated strategy can (but doesn't have to) have fixed SL/PT. It is advisable to use different exit rule if you uncheck this setting, for example exit after X bars, otherwise the strategy will have no way to exit the trade.

4.2.2 Parts to improve

This setting is visible only if you chose that you want to improve an existing strategy in What to build setting.



It simply allows you to choose which parts of the strategy you want to improve. You can choose to improve Entry rule, Exit rule or Order type, and you can choose one of the three options:

Add – add new conditions to the existing ones

Replace – delete the existing conditions and generates new ones

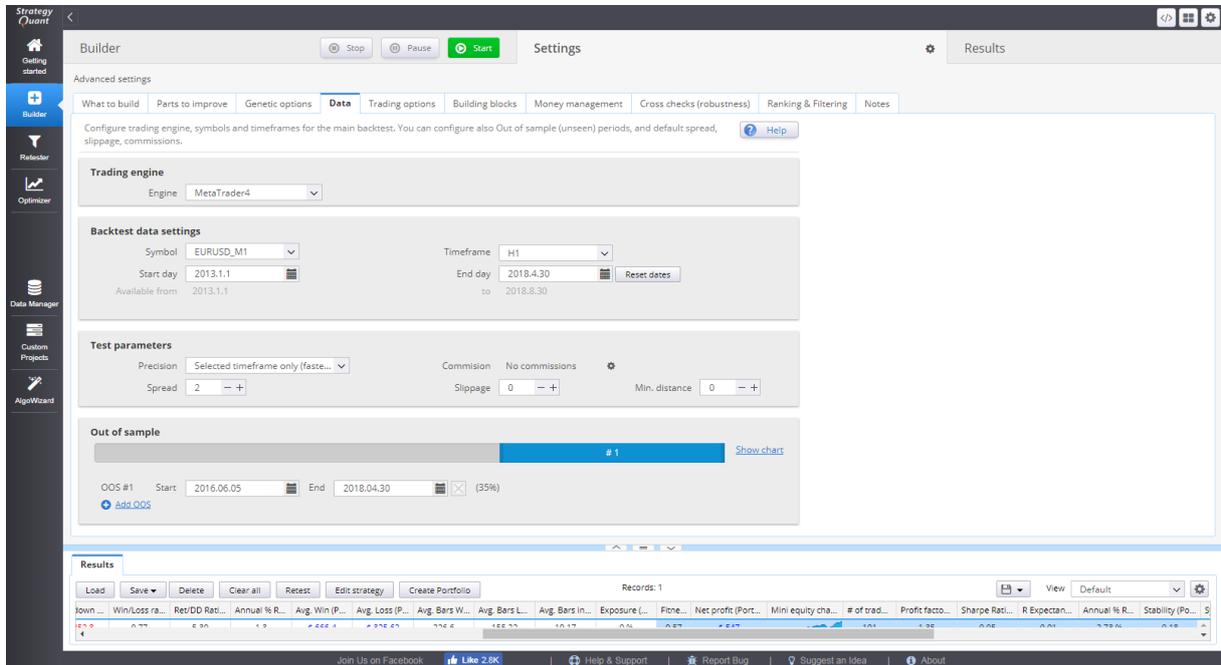
Add or Replace – randomly decides whether to use Add or Replace.

Strategy generator in this mode will simply take your strategy, tries to recognize what are the entry and exit rules, and then depending on your configuration generates additional conditions or replaces for of the conditions in your existing strategy with new ones.

Note that there is another more flexible way to improve your strategies – using Build from strategy template. In this case you'll have to open your existing strategy in Wizard, and put a generation placeholders to the parts you want to change. Then start build with generation using this template and it will produce new strategies that are based on your original strategy template, but with placeholders replaced with newly generated conditions or actions.

4.2.3 Data

This is where you can configure the symbol, timeframe, and time range on which the strategies will be backtested.



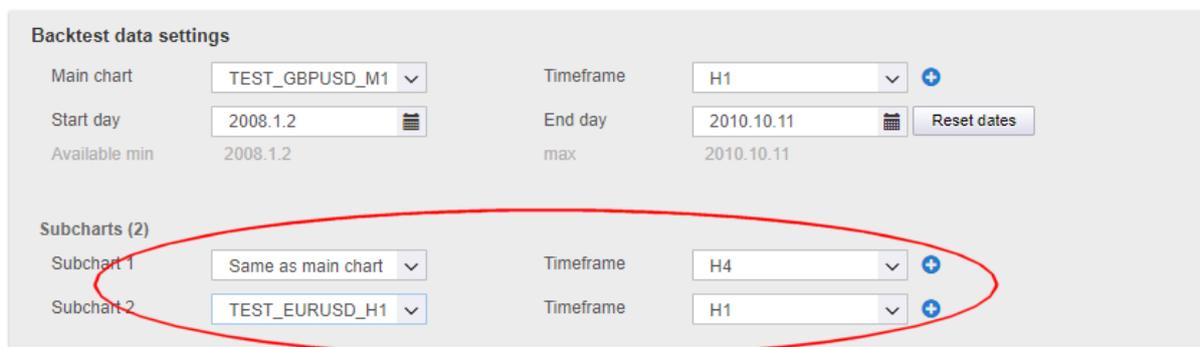
Trading Engine

allows you to choose which of the available backtesting engines (MetaTrader4 / MetaTrader 5 etc.) should be used for testing the strategy.

Backtest data settings

choose the symbol, timeframe and date range on which the strategies will be tested.

If you use multi-TF or multi-symbol strategies you must select symbol and timeframe for each additional chart:



Test precision

testing precision means how the price is data simulated during the backtest. It is usually sufficient to use Selected Timeframe mode for Market orders, or Tick simulation mode for Stop/Limit orders. You can also use faster mode for generation, and then slower mode for strategies retest.

Out of Sample Period

Allows you to split the history data to two parts:

- **In Sample** - it is used during genetic evolution to compute the rank of the strategies, so that the program is able to compare them
- **Out of Sample** - this part of the data is used to verify if strategy really works as expected also on data that it was not evolved

Test parameters

Testing precision

means how the price is data simulated during the backtest.

It is usually sufficient to use Selected Timeframe mode for Market orders, or Tick simulation mode for Stop/Limit orders.

You can also use faster mode for generation, and then slower mode for strategies retest.

Selected Timeframe only

it is the fastest testing mode. It uses only the main timeframe to simulate the prices, it creates four “ticks” – at bar Open, High, Low and Close values.

This results in a very fast backtesting with acceptable accuracy for a quick preview. However, for Stop or Limit orders the testing accuracy might not be sufficient, and you should try more precise mode.

1 Minute data

slower testing mode, it uses minute data (if available) to simulate price changes during the testing. It creates 4 ticks every minute, and thus simulates also movement within the bar.

Real Tick – custom spread

this mode uses Bid value from real tick data (if available) for exact price simulation, but it computes Ask value using the custom spread you specify. It is slower than the other modes and it should be used for final verification of new strategies.

Real Tick – real spread

this mode uses real tick data for exact price simulation with real Bid and Ask prices. It is much slower than the other modes and it should be used for final verification of new strategies.

Spread

You can specify spread used in backtest – it is a difference between Bid and Ask price.

Other (Slippage, Min Distance, Commission)

clicking on this link will open a new popup dialog, where you can configure these additional settings.

Other parameters ✕

Slippage – + in pips Min. distance – + in pips

Minimum distance of stop/limit order from current market price. It can be different for every broker. Because of this, some orders that are placed too close to the market price will be not accepted.

Commission

None
 Percentage based
 Size based

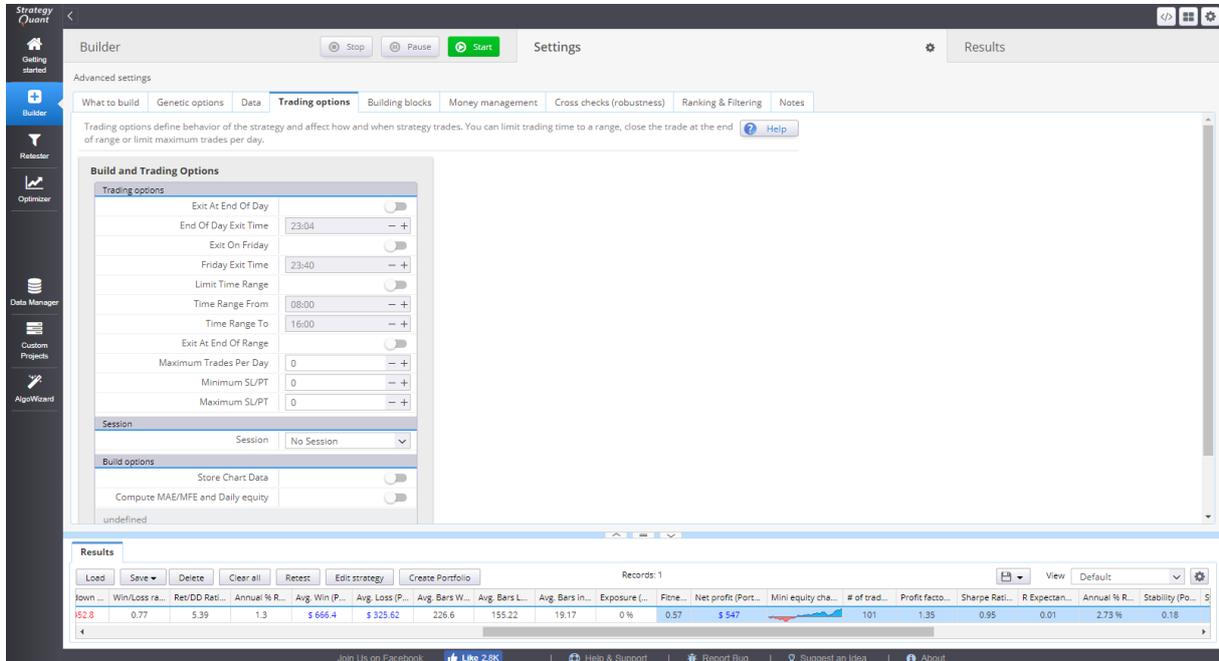
Default	
Commission	<input type="text" value="0"/> – +

Simple size-based commissions
Simple method of commissions computation - it uses commissions in \$ per lot and multiplies it by actual trade size.

[Close](#)

4.2.4 Trading options

These settings allow you to specify the properties of the generated strategies as well as testing conditions. Trading options govern behavior of the strategy in backtesting engine – for example whether all orders should be closed at the end of day, etc.



Exit at end of day/end of range

if selected, the strategy will close all position at the end of the day or end of the trading range (if defined). This way you'll have no position open overnight.

Limit Signals to Time Range, Time Range From, Time Range To

this limits the hours the strategy is checking for entry signal to a given time range.

If used in combination with **Exit At End Of Range** then all open positions are closed at the end of the range.

If you don't check the **Exit At End Of Range** then the strategy will not open new trades outside the trading range, but the already opened positions will be not closed.

Exit At End Of Range, Exit At End Of Day, Exit On Friday

three settings that manage whether trades should be closed at the end of range, day or week.

Maximum Trades Per Day

you can limit maximum trades the strategy takes per day

Store Chart Data

if checked SQ X will save also complete chart data for the backtest, so you will be able to see history chart with backtest with used all indicators and all orders drawn on chart.

This can help you review the strategy and see visually how it is trading.

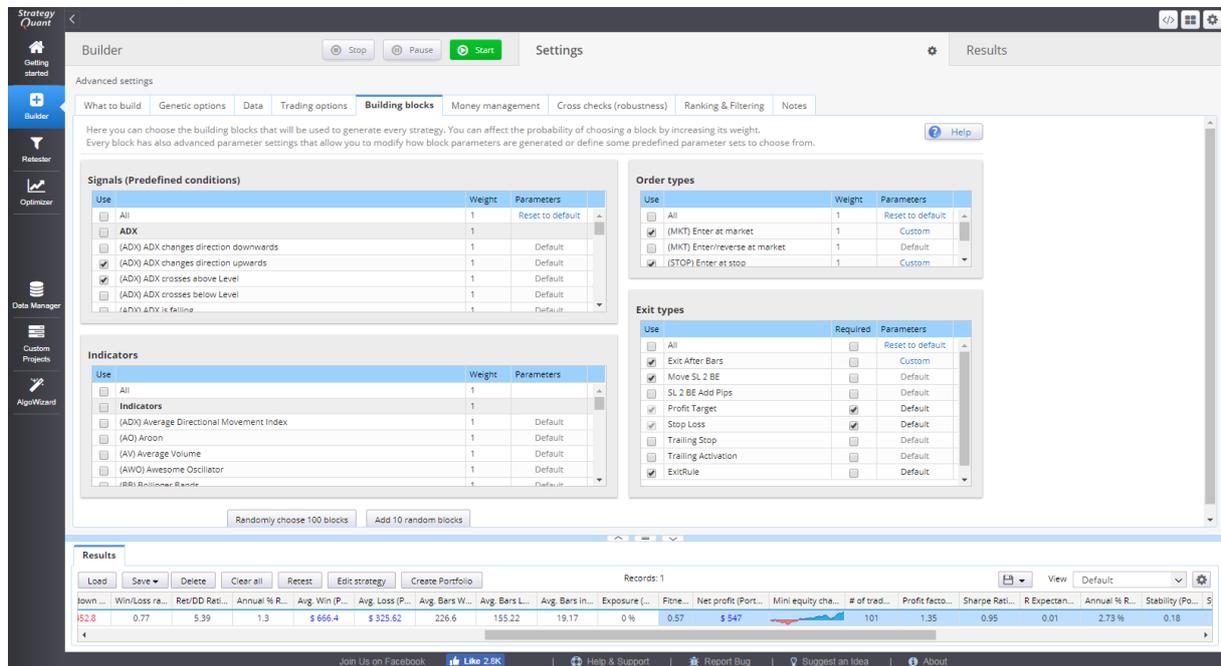
Note that storing chart data will take some time, so backtest will be slower and produced strategy file will be significantly bigger.

Compute MAE/MFE and Daily Equity

this is used to compute MAE/MF and equity values. If not used the generation speed can be increased by 10 - 20 %.

4.2.5 Building blocks

Building blocks are the core components that are combined to create rules and actions for every trading strategy.



There are three separate panels:

Entry Rules Building Blocks

Entry building blocks can be divided into several parts:

- **Indicators** -RSI, CCI, Momentum, etc.
- **Price ranges** – indicators like ATR, Bar Range
- **Price values** – simple price values like Open, High, Low, Close
- **Operators** - that are used to compare and combine the rules, for example <, >, and, or, etc.
- **Simple rules** -predefined rules like “CCI is rising”, “CCI is higher than level”, etc.
- **Other** – other blocks, like Number constant, Hour, Minute of day, Day of week

Order and Exit Types

These are blocks that can be used in action – what kind of order to open (Market, Stop, Limit) and how to exit the order – Stop Loss, Profit Target, Trailing Stop, etc.

You can find description of all Entry and exit types.

You can choose each block that you want to use by checking its checkbox. You can also optionally increase its weight or %. Weight is used for blocks that compete with each other – block with bigger weight has bigger chance to be chosen.

Percentage is used for exits – how big probability is to use given exit method.

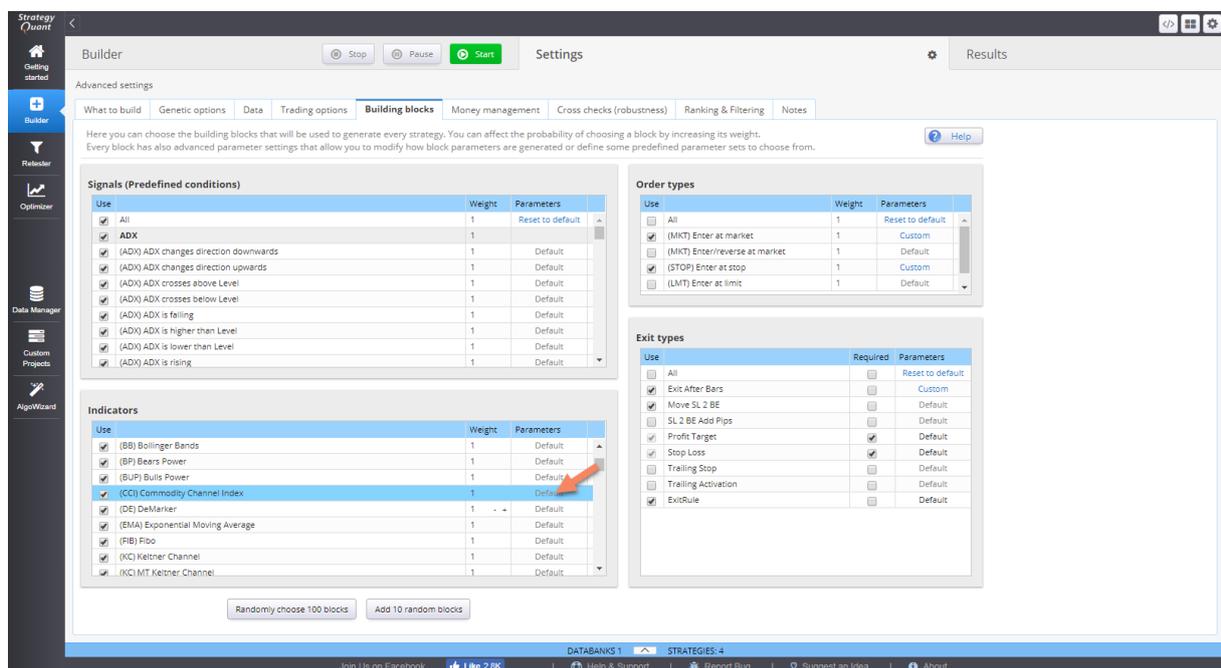
This way you can choose blocks want to use in the strategy, so you can select your favorite indicators, or choose for example only price data + operators if you want to generate strategies based only on price.

Good practice

According to our experience, you can sometimes get better results if you don't check all the available components, but narrow your choice to a smaller group of indicators or price values.

Advanced - Block parameters

There is a possibility to edit parameters for every generated block. By default, it uses Default parameters, you can click on the [Default](#) link next to the block to modify its config.



Edit block (CCI) Commodity Channel Index
✕

Parameter sets

Parameter values

Parameters of this building block. They can be generated randomly or have fixed value - you can configure it for every parameter by double-clicking on it in the grid below.

If you wish SQ to choose from some most common values (for example for periods), define them as parameter sets in the **Predefined Parameter Sets** tab.

Parameter name	Type	Generation	Possible values
Chart	data	random	All charts
Computed From	int	random	Close,Open,High,Low,Median,Typical,Weighted
Period	int	random	min: Min period, max: Max period, step: 1
Shift	int	random	min: Min shift, max: Max shift, step: 1

[Reset to default](#)
[Close](#)
Save

Let's see configuration on Parameters tab for CCI. In the screenshot above you can see list of all parameters used in CCI building block:

- **Chart** – every indicator is computed on a given chart. If you use only one chart (simple strategy) it will be computed from data on main chart
- **Computed From** – price data from which CCI is computed
- **Period** – CCI period
- **Shift** – CCI shift value

Every indicator or other building block has its own set of parameters, most of them contain Chart and Shift, plus some number of indicator parameters like periods.

Parameters

You can choose the exact range for random generated value, or set some of the parameters fixed. To edit a parameter, double click on it in the grid.

Edit parameter [X]

Parameter: Period

Type: **Randomly generated** Fixed value

Minimum: **Global** Custom 15

Maximum: **Global** Custom 80

Global Minimum and Maximum can be configured in
What to build -> Additional build config

Step: 1 - +

[Close](#) **Save**

Then you can choose if the parameter will have fixed value or if it will be randomly generated. For random parameters you can also choose list of values to choose from, or range from-to for numeric parameters.

Parameter Sets

There is one more possibility to define block parameter ranges. You can create multiple sets of parameters, where in each set some of the parameters could be generated randomly, and other have fixed values.

Edit block (CCI) Commodity Channel Index ✕

Parameter sets

Parameter values

Parameters are either generated all randomly as defined in tab **Parameters**, or SQ can choose one of the predefined parameter sets specified below.
 This allows you to define most common indicator values (for example periods). Even in parameter sets some of the parameters can be generated randomly (denoted by *), while others will have predefined value.

Set name	Possible values	Weight		
All random	Chart=*,Computed From=*,Period=*,Shift=*	1		
Default set 1	Chart=*,Computed From=*,Period=14,Shift=*	1	✕	
Default set 2	Chart=*,Computed From=*,Period=20,Shift=*	1	✕	
Default set 3	Chart=*,Computed From=*,Period=30,Shift=*	1	✕	
Default set 4	Chart=*,Computed From=*,Period=40,Shift=*	1	✕	
Default set 5	Chart=*,Computed From=*,Period=50,Shift=*	1	✕	
Default set 6	Chart=*,Computed From=Close,Period=14,Shift=*	1	✕	
Default set 7	Chart=*,Computed From=Close,Period=20,Shift=*	1	✕	
Default set 8	Chart=*,Computed From=Close,Period=30,Shift=*	1	✕	

[Reset to default](#) [Close](#) Save

So instead of generating (for example) CCI Period randomly from range between 10 and 50, you can create a few parameter sets, and let SQ randomly chose CCI period only from values 14 and 30.

Parameter sets compete with normal parameter definitions for which of them will be actually used in the generated block. You can use Weight to increase probability that a particular set is used.

For example, by setting Weight of Parameters to 0, while keeping Weights of parameter sets to 1 you tell SQ to generate parameters only using parameter sets.

4.2.6 Money management

Money management (or position sizing) specifies how many lots or shares are traded on each trade.

StrategyQuant contains flexible money management options that can be used in the program and later also in real trading in MetaTrader.

Brief description of different money management types:

Fixed size

strategy will trade with fixed number of lots. This is recommended setting when you generate new strategy, because it gives you clear overview of strategy real performance.

Fixed amount

strategy will risk a fixed amount of money for every trade.

This is basic money management without compounding. It can be used to test real performance of strategies where Stop Loss is based on volatility (ATR), or if you want to compare the performance of strategies with different Stop Loss.

Risk fixed % of account

advanced money management that is recommended for real trading. The strategy will risk a given % of equity on every trade.

This is simple, but very effective money management that will allow the strategy to increase number of lots as your account grows.

It is generally recommended to risk maximum 2-5% of account equity per one trade.

Risk fixed % of balance

Like Risk fixed % of account, but it uses account balance instead of equity.

Account balance illustrates your closed trades Profit/Loss, while Equity is the real-time calculation of Profit/Loss, considering both open and closed. Positions.

Stocks size by price

This position sizing method is used especially used for stocks. **Use account balance** forces SQ to use the current account balance instead of the initial capital set

4.2.7 Cross checks (robustness tests)

Cross check tests are additional tests that can be applied to your strategy. The idea behind robustness testing is to verify how well the strategy will behave when there are small changes in inputs, history data or other components of the strategy.

This section is described in more detail in the Advanced functionality and Cross checks - robustness tests and analysis of this guide.

4.2.8 Ranking options

When the strategies are generated, every new strategy is backtested on a history data and the results of the backtest are then used to compute the *Fitness* (rank) of the strategy. Fitness is number from 0 to 1 and it should reflect the "quality" of the strategy according to the given criteria.

In this screen you can configure:

- how this *Fitness* value is computed (Strategy Selection criteria)
- how many top strategies are stored in the databank
- which strategies are stored to databank and which are thrown away (Custom conditions)

Databank Options

The best strategies found are continually stored into the **Databank**.

It is not possible to store every strategy (remember that StrategyQuant can create thousands of new strategies every hour) so we must specify how many strategies should be stored in the databank, how they should be sorted to find out the best ones and which strategies should be thrown away.

Maximum Strategies to Store in Databank

simply the maximum number of strategies remembered in the program

Stop generation when

allows you to define whether the generation process will be terminated when databanks if full or will continue until stopped manually.

Strategy Quality ranking (fitness)

Here you can choose from which data the strategy criteria will be computed and what criteria should be used to compute the total Fitness of the strategy.

Use (main data backtest / cross checks):

You can select if you want to compute fitness (strategy fitness score) from main data or from portfolio (if you use also additional data).

It is generally recommended to use Portfolio, then the strategy fitness will be computed from the sum of all trades across all the symbols in your portfolio.

If you use this setting and you don't have any additional data, fitness will be automatically computed from main data.

Determine best strategies by

You can choose from the predefined most used criteria, or you can build your own complex fitness function based on multiple criteria, each with a different weight.

Note that if you choose to combine too many criteria, they might "fight" against each other without achieving what you expected. **Profit Factor** criterion to maximize Profit Factor of a strategy.

You can find explanation for every Ranking criteria

Strategy filtering conditions

you can define rules to throw away strategies with bad properties. For example, it usually makes sense to throw away strategies that result in loss or these that have too little number of trades.

Automatic filters

automatic filters should be used to dismiss strategies that are obviously wrong such as 0 trades strategies, strategies having no profit or strategies which have no filled orders. All the available options can be seen below.

Configure automatic filter ✕

Automatic filter should be used to dismiss strategies that are obviously wrong, for example they don't produce any trades, or trades are ambiguous and cannot be backtested properly.
It is highly recommended to keep all automatic filters turned on!

Threshold values for these conditions are: at least 10% or max 100 of the trades must be bad in order to mark strategy as bad.
For ambiguous trades the threshold is decreased to 5% or max 50 trades.

Dismiss strategies with these problems:

<input type="checkbox"/> Select all / Deselect all	<input checked="" type="checkbox"/> zero PL trades
<input checked="" type="checkbox"/> no trades	<input checked="" type="checkbox"/> zero duration trades
<input checked="" type="checkbox"/> too many ambiguous trades	<input checked="" type="checkbox"/> unfinished trades
<input checked="" type="checkbox"/> too many open trades	<input checked="" type="checkbox"/> too little trades
<input checked="" type="checkbox"/> no filled trades	<input checked="" type="checkbox"/> too many trades closing at the same bar
<input checked="" type="checkbox"/> outlier trade - one exceptionally big trade	

[Close](#)

Custom filters

allow you to specify your own custom rules that will be evaluated for every strategy. If the strategy matches any of the rules, it will be dismissed.

This is useful to quickly dismissing strategies that result in loss or with too little number of trades.

There are separate values for In Sample and Out of Sample performance, and for Robustness Tests (if used) or Portfolio (if you used additional data).

Strategy filtering conditions
 Configure which strategies will be saved into Results databank

Automatic filters
[Configure automatic dismissal](#)

Custom filters

<input type="checkbox"/>	Left value	<>=	Right value	
<input checked="" type="checkbox"/>	Net profit (IS)	> ▼	500	×
<input checked="" type="checkbox"/>	# of trades (IS)	> ▼	200	×

The abbreviations used in custom conditions are:

- **IS** (In Sample) - result of In Sample part of data
- **OOS** (Out of Sample) - result of Out of Sample part of data
- **RT** (Robustness Tests) - result of robustness tests
- **P** (Portfolio) - results for portfolio (if exists)

Edit condition : this > 500 ✕

Net profit ▾

From backtest Main data ▾

Sample type Full sample In sample Out of sample

Direction Both Long Short

Result in Money Percent Pips

Apply percentage ratio

[Close](#) Save

Results can be evaluated using **Money**, **Percent** or **Pips** values for either **long**, **short** or **both** directions.

4.2.9 Genetic options

when you set Build mode to **Genetic evolution** in **What to build -> Build Mode**, a new tab **Genetic options** will appear in the settings.

The screenshot displays the 'Genetic options' settings in the StrategyQuant X Builder. The interface is organized into several panels:

- Genetic options:**
 - Max # of Generations: 10
 - Population Size (per island): 5
 - Crossover Probability: 50%
 - Mutation Probability: 20%
- Islands options:**
 - Islands (separate evolution): 10
 - Migrate every Xth generation, X =: 50
 - Population migration rate: 20%
- Initial population generation:**
 - Initial population size required: 50
 - Use strategies from Initial population databank as evolution start
 - Generated decimation coefficient: 1
- Filter generated initial population:**
 - Only strategies that fulfill the conditions below will be accepted
 - Table with conditions:

	Left value	<=>	Right value	
<input checked="" type="checkbox"/>	# of trades	>	1000	x
<input checked="" type="checkbox"/>	Net profit (IS)	>	-5000	x
 - Add conditions button
- Evolution management:**
 - Start again when finished (continuous repeating evolution)
 - Restart evolution if fitness of 'In sample' stagnates for 3 generation(s)

There you can set up various properties of genetic evolution.

Genetic options

Max # of Generations

number of generations for which the population will be evolved. Recommended value from 5 – 100. It usually doesn't bring much improvement to use too many generations, it is better to just restart the evolution and start from the scratch.

Population size

size of population on one island. Recommended value from 10 – 100 or even more. Please note that if you use more than one island, your total population will be (number of islands) x (population size), so make sure you'll not have some very extreme number.

Crossover and Mutation probability

probability of basic genetic operations. You can experiment with these values, for example increasing mutation should generate more diverse strategies.

Island options

Islands

number of separate islands. Islands are a new concept in SQ X, they allow running genetic evolution separately in isolated islands, with occasional migration of individuals between islands.

There's no problem having just one island, the recommended value is 1-10. It doesn't make much sense to use more than 10 islands, it could make your total population very big and it would take a lot of time to just evolve one generation.

Migrate every Xth generation

how often to migrate some individuals from island to island. It is usually good to migrate the individuals, it can "unlock" some island that got stuck in the local minimum. It shouldn't be too often, because then we'd lose the diversity of the independent islands.

Recommended value is for example every 10 generations.

Population migration rate

how many strategies in the population will be migrated. It should be something like 1-5 strategies, depending on your population size, so for population size=10 use value like 10-20%, for population size=100 use value like 1-5%.

Initial population generation

Genetic evolution starts from some initial generation. It will either be generated randomly, or you could use some existing strategies as initial population and try to improve upon them.

Use strategies from Initial population databank as evolution start

checking this will show you one more databank "Initial population" where you can load your existing strategies. These strategies will then be used as initial population. If there is not enough of them, the rest will be generated randomly.

Please note that initial population from databank is NOT filtered using Initial population filter.

Generated decimation coefficient

Decimation means that there will be X-times more strategies (that pass filters) generated than required, and from these the best will be chosen.

If you set decimation for example to 3, it will generate 3x more strategies for initial population and choose the best from this.

Using decimation will improve quality of initial population, but it will take much more time to generate it.

Please use it wisely, because it can greatly increase the number of strategies that need to be generated for initial population, so it might take very long time to just generate the initial strategies before evolution even starts!

Filter generated initial population

Here you can set up filter to set some basic minimum that a strategy should have to be in the initial generation.

Please use it wisely, genetic evolution should be able to improve any population of strategies, so don't be too strict with your initial generation.

The only recommended filter to use is by number of trades – to filter out strategies that are not trading at all.

Evolution management

Start again when finished

if checked it will restart the building process when it finished. So you can let it run autonomously, and SQ will evolve more and more populations until you stop it.

Restart evolution fitness if..

it will restart evolution if we are stagnating in fitness – this means that the population as a whole is not improving, so it is better to start again from the beginning.

4.3 Results

4.3.1 Databanks

Databank is a storage place where the generated/tested strategies are stored with their results.

The screenshot shows the 'Results' window with a table of strategy performance. The table has columns for Strategy Name, Fitness, Symbol, TimeFrame, Net profit, Mini equity chart, # of trades, Profit factor, Sharpe Ratio, R Expectation, Annual % Return, Stability, Symmetry, Drawdown, Win/Loss ratio, and Return/DD Ratio. Three strategies are listed: BonusStrategy1, Strategy 0.98, and Strategy 0.155.

Strategy Name	Fitness	Symbol (Portfolio, IS)	TimeFrame (Po...	Net profit (Port...	Mini equity cha...	# of trad...	Profit facto...	Sharpe Rati...	R Expectan...	Annual % R...	Stability (Po...	Symmetry (...	Drawdown ...	Win/Loss ra...	Ret/DD Rat
BonusStrategy1	0.84	EURUSD_M1	H1	\$ 2441.6		232	1.57	4.24	0.03	5.57 %	0.42	18.75 %	\$ -452.8	0.77	5.39
Strategy 0.98	0.47	EURUSD_M1	H1	\$ 1132.09		313	1.11	0.48	0.03	3.04 %	0.56	84.08 %	\$ 2401.19	1.28	0.47
Strategy 0.155	0.63	EURUSD_M1	H1	\$ 662.5		263	1.17	0.35	0.08	1.67 %	0.79	0 %	\$ 570.2	0.87	1.15

Each mode (Build/Test/Improve/Optimize) has its own independent databank table.

For memory reasons Databank cannot keep unlimited number of strategies, instead it stores the selected number of top strategies, for example top 100 or top 1000 strategies.

The configuration how many strategies to store in the Databank and how to sort them can be set in the Settings -> Ranking options

Databank functions

Every strategy result in the Databank can be viewed in the Results screen by using double-click.

Load

this button is available only in **Retest** databank, it loads Strategy projects (.str files) so they can be retested.

Save (as Strategy/Source code)

this way you can save your strategy, you should always save good strategies to a Strategy Project (.str) so that you can work with them later.

Edit strategy (Parameters/Trading rules)

here you can manually edit the selected strategy - either the values of its parameters or the trading rules.

Delete/Clear all

this will delete selected or all strategies from the list

Create portfolio

it will combine selected strategies to a portfolio - a special report that combines trading results of multiple strategies. You can then check/analyze the results of whole portfolio of strategies.

Views

Each row in the table represents one strategy plus, the results of the strategy backtest - number of trades, total profit/loss, Fitness score, etc. are visible in the table columns.

Results are divided into In Sample and Out of Sample parts.

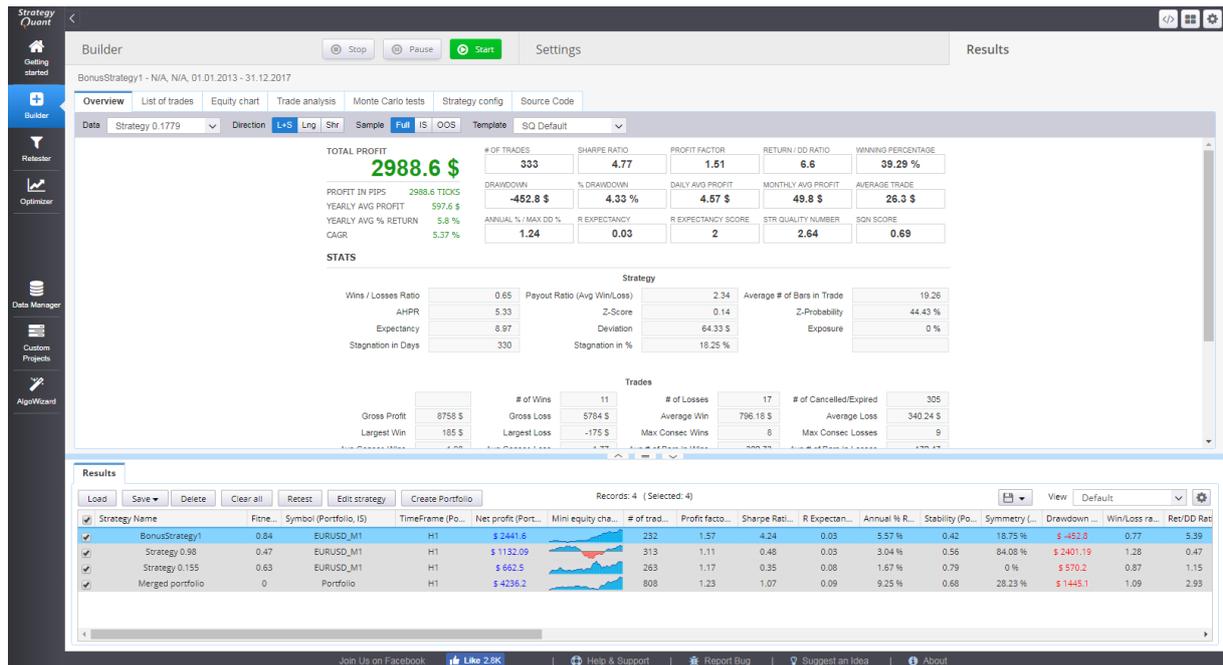
All the columns in the databank are customizable, you can choose which columns you want to see using views.

Export databank contents to CSV/XLS

you can also export the whole table content to a CSV or XLS file

4.3.2 Overview

Overview screen displays various statistics computed from the backtest data.



4.3.3 List of trades

Contains complete list of trades generated by the backtest.

If you backtested your strategy on multiple symbols/timeframes, you can switch between the results or check the list of trades for whole portfolio.

4.3.4 Equity chart

Displays equity chart of the strategy.

Stagnation marker shows the maximum stagnation period - this is the longest period that it took strategy to make new high on the equity.

In case of portfolio, it displays charts for every symbol/timeframe tested, plus equity of complete portfolio.

4.3.5 Trade analysis

This screen displays yearly performance of the strategy plus configurable small charts with trades per hour, day of week, day of month, etc.

4.3.6 Robustness tests - Monte Carlo

This screen is empty if you didn't configure any Robustness Tests.

Otherwise it contains equity chart for every robustness test simulation and table of important confidence levels computed using Monte Carlo analysis of simulations.

For information about how to use Robustness tests and how to interpret the results, check the Advanced functionality and Cross checks - robustness tests and analysis section of this guide.

4.3.7 Strategy settings

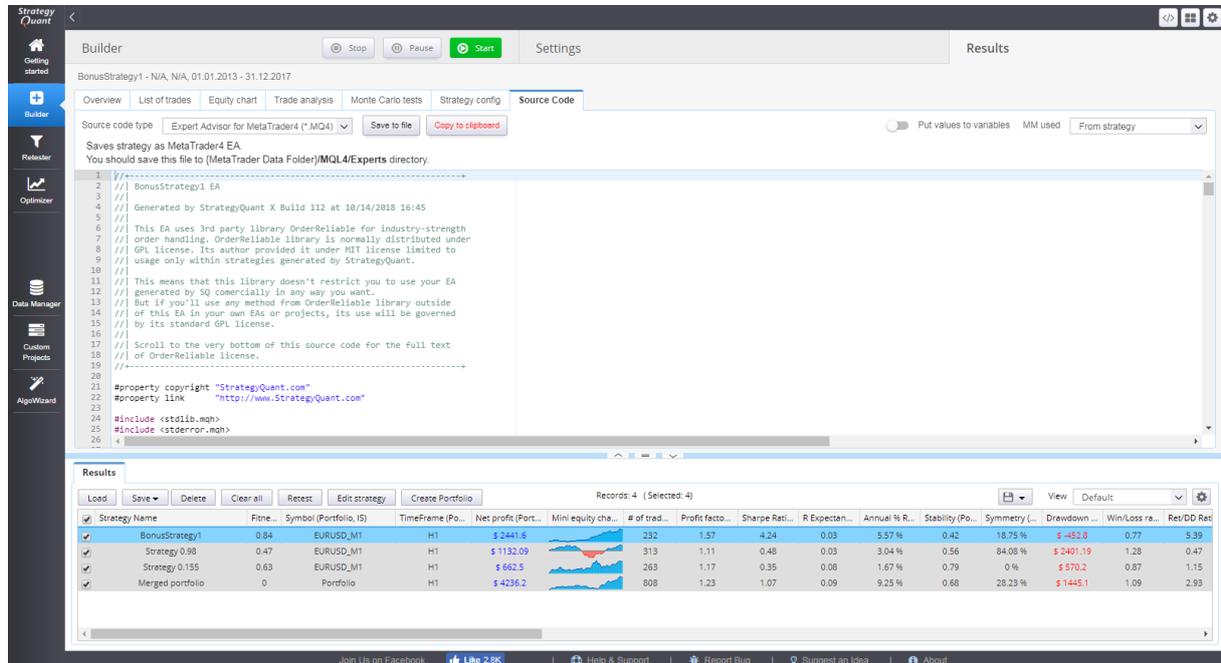
This screen displays current settings of the program versus settings that were used when the strategy was last time backtested/generated.

The differences are highlighted with **red color**.

To quickly load settings from the strategy you can use the button **Apply Strategy Settings**. This will load settings from the strategy and applies it to the selected program section.

4.3.8 Source code

Source code tab is where you'll get the result of the program. For every selected strategy it generates selected source code.



Source Code Type

There are three types of source code that can be generated:

Strategy XML

allows you to see the data structure in XML data format

Pseudo Code

human readable pseudo code of the strategy. You can see the strategy logic and you can use it for manual trading.

MetaTrader4 / MetaTrader5 Expert Advisor

the EA code for MetaTrader. You can save the EA code to MetaTrader/experts directory and you'll be able to run your new EA in MetaTrader.

NinjaTrader code

Support for NinjaTrader will be added later.

EasyLanguage code for Tradestation/MultiCharts

the code for TradeStation platform. It can be also compiled in MutliCharts trading platform

Put values to variable

by checking this checkbox, you can generate code where all the number constants like indicator periods, comparison constants etc.

will be parameterized - a parameter will be added for every constant.

This allows you to optimize your EA in MetaTrader, or test it with different parameter values.

MM used

this option allows you to include money-management settings from builder/retester/optimizer into the strategy source code.

5 Advanced functionality

5.1 Cross checks - robustness tests and analysis

Overfitting or curve-fitting strategy to the historical data on which it was build is the biggest danger of strategies generated using any machine learning process.

During or after developing new strategy you should make sure your strategy is robust - which should increase the probability that it will work also in the future.

What is robustness?

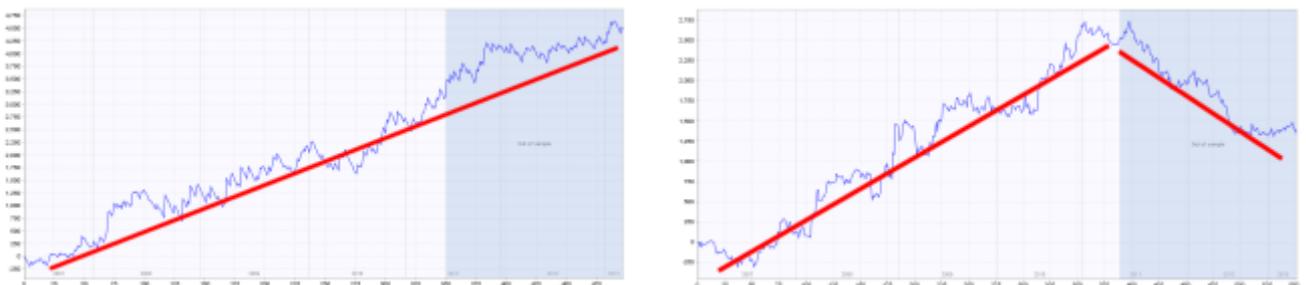
It is simply the property of strategy of being able to cope with changing conditions:

- First of all, strategy should work on unknown data (if the market characteristics didn't change) either with or without periodic parameter re-optimization
- It should not break apart if some trades are missed.
- Robust strategy should not be too sensitive to input parameters - it should work even if you slightly change the input parameter values, such as indicator period or some constant or if historical data are slightly changed – spread or slippage is increased, and so on

How to test strategies for robustness

1. Make use of In Sample and Out of Sample periods

the most basic test for robustness is testing the strategy on Out of Sample data. If you run genetic evolution, the strategy is evolved only on the In Sample part of data. The Out of Sample part is "unknown" to the strategy, so it can be used to determine if the strategy performs also on unknown part of data.



The blue part of each chart is the Out of Sample (unknown) data., We can see that the strategy on the left performs well also on this part, while strategy on the right fails on the unknown data - it is almost certain to be curve fitted.

2. Use Cross checks build in Builder and Retester

StrategyQuant X allows you to use cross checks (robustness tests) during the strategy build, or when retesting the strategies.

There is a number of cross checks that can be used, ranging from simple ones to very complex ones, and they are simple to choose – you can just move the slider in the simple settings:

Cross checks (strategy robustness tests)

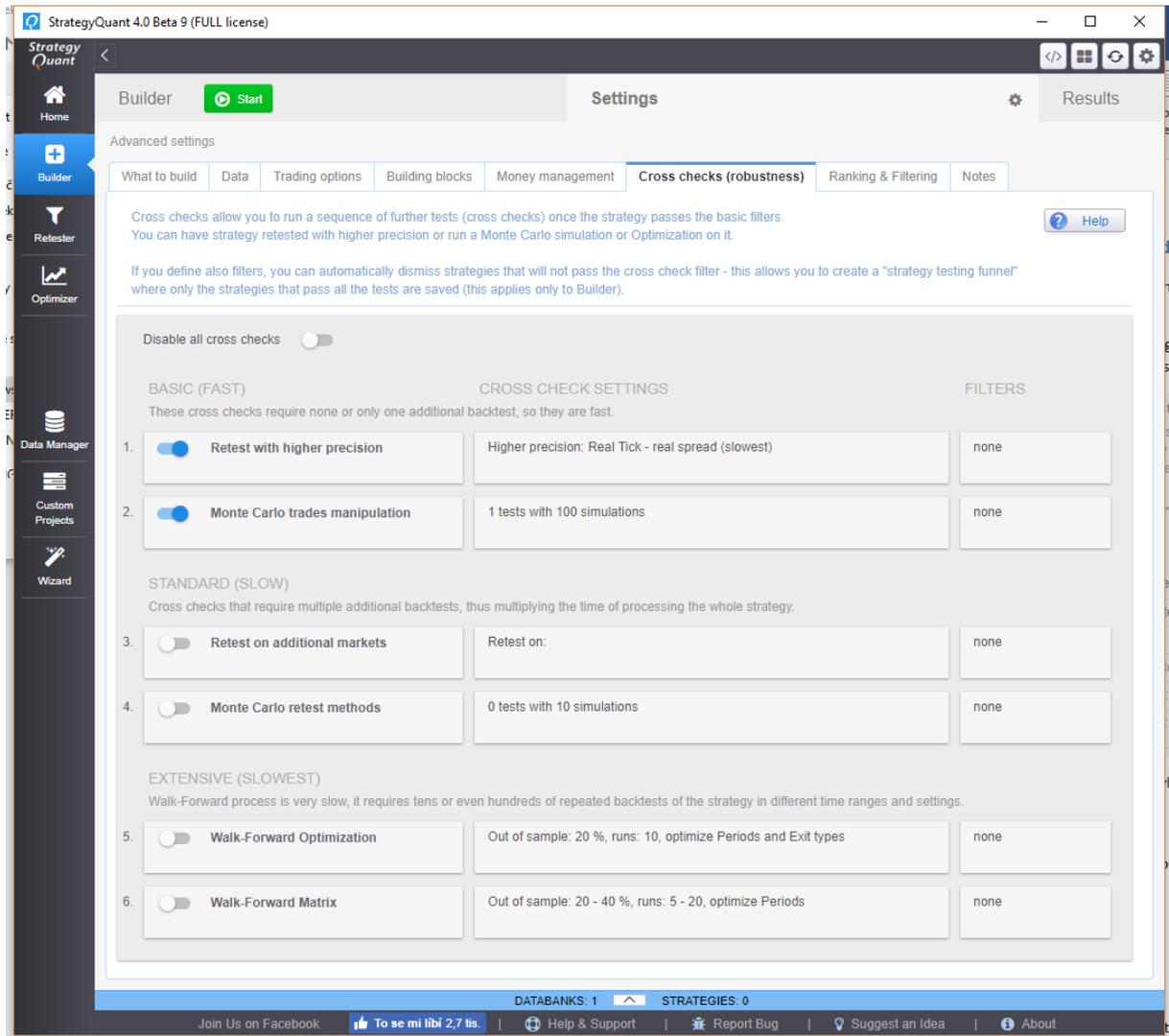
Cross checks perform additional backtests on the generated strategy, they can make generation of one strategy much slower, but they can be used to automatically filter out "bad" strategies. [Learn more](#)



None Basic Standard Extensive

<input checked="" type="checkbox"/> Retest with higher precision	Real Tick - real spread (slowest)
<input checked="" type="checkbox"/> Monte Carlo trades manipulation	1 tests with 100 simulations
<input type="checkbox"/> Retest on additional markets	
<input type="checkbox"/> Monte Carlo retest methods	0 tests with 10 simulations
<input type="checkbox"/> Walk-Forward Optimization	OOS: 20 %, runs: 10
<input type="checkbox"/> Walk-Forward Matrix	OOS: 20 - 40 %, runs: 5 - 20

or turn them on/off individually in advanced settings, where you can also change their configuration and filters:



The predefined configurations for every cross check should be optimal settings, a balance between speed and thoroughness.

5.1.1 Retest with higher precision

This test is simple – it backtests the strategy again on the same data, but with higher precision.

It is usually best to make the main test on the fastest Selected timeframe precision, because it can very quickly filter out bad strategies – these that produce no trades or whose Net profit is negative.

Once we have strategy that is profitable, it can be automatically retested with higher precision using this cross check – ensuring that the strategy works also when tested with better precision.

It can happen that strategy that is profitable with Selected timeframe precision ceases to be profitable when retested with higher precision – it is because the lower precision simply is less exact.

5.1.2 Monte Carlo trades manipulation

This cross check run simulations where in each simulation it manipulates the existing trades – shuffles them, misses some and so on.

It is very quick, because it doesn't require running backtests, it works on already existing trades from main backtest.

The idea behind this is to verify how much the strategy equity curve depends on particular order of the trades, and what would happen if some trades are missed.

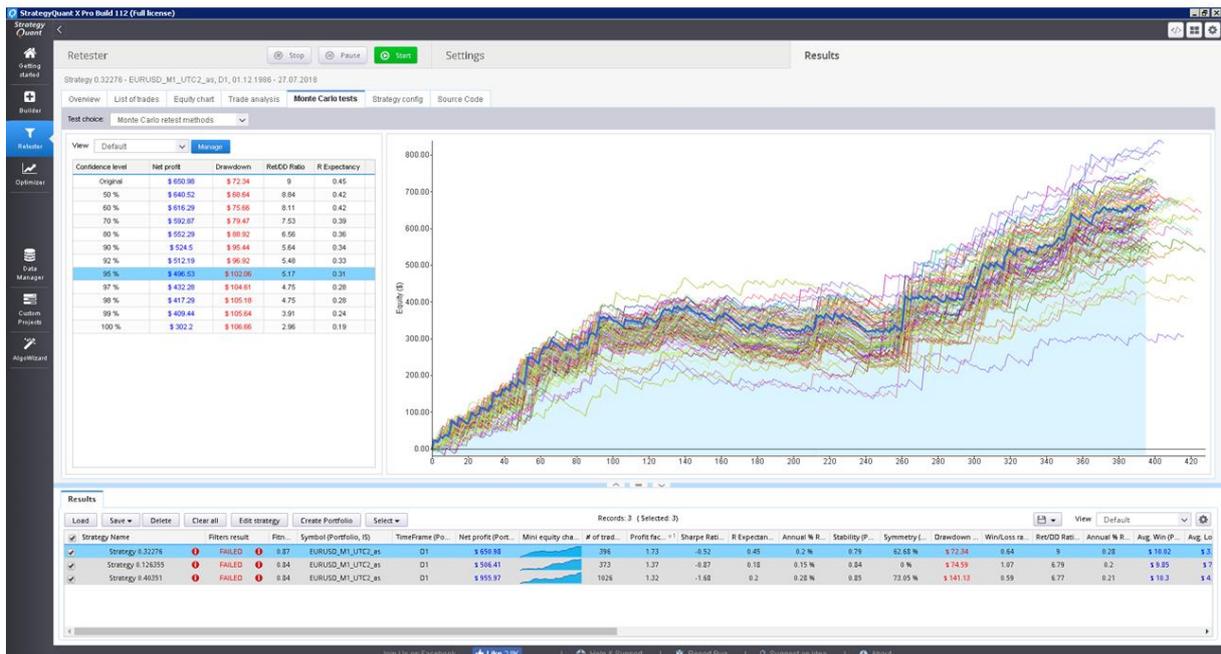
You can do these trade manipulations in every simulation:

Randomize Trades Order - this is the simplest test, it randomly shuffles order of the trades. This doesn't change the resulting Net Profit, but it is very useful in examining different variations of Drawdown that can be a result of different order of trades.

Randomly Skip Trades - it will randomly skip trades with given probability. In real trading you can often miss a trade because of platform or Internet failure, or simply because you paused trading for some time. This test will give you an idea how the equity curve might look like if some trades are randomly skipped.

Interpreting the results

Robustness tests output the results as a set of equity charts for each testing run AND a table showing the results of Monte Carlo simulation.



In this example we've run 100 simulations, with randomly skipped trades. We can see what would be the equity for each of these simulations and the table on the left provides us the valuable information on the strategy properties during these simulations.

Test choice: Monte Carlo retest methods

View: Default Manage

Confidence level	Net profit	Drawdown	Ret/DD Ratio	R Expectancy
Original	\$ 650.98	\$ 72.34	9	0.45
50 %	\$ 640.52	\$ 68.64	8.84	0.42
60 %	\$ 616.29	\$ 75.66	8.11	0.42
70 %	\$ 592.87	\$ 79.47	7.53	0.39
80 %	\$ 552.29	\$ 88.92	6.56	0.36
90 %	\$ 524.5	\$ 95.44	5.64	0.34
92 %	\$ 512.19	\$ 96.92	5.48	0.33
95 %	\$ 496.53	\$ 102.06	5.17	0.31
97 %	\$ 432.28	\$ 104.61	4.75	0.28
98 %	\$ 417.29	\$ 105.18	4.75	0.28
99 %	\$ 409.44	\$ 105.64	3.91	0.24
100 %	\$ 302.2	\$ 106.66	2.96	0.19

What do these values mean?

The first row displays values of Net Profit, Maximum % Drawdown etc. of original strategy for comparison.

The rest of the rows display values at different confidence levels.

These numbers are a result of Monte Carlo analysis applied on our 10 random simulations.

For example, values at 80% confidence level mean that there is 20% chance that Net Profit, Drawdown etc. will be worse than the confidence level values.

Values at 90% confidence level mean that there is 10% chance that Net Profit, Drawdown etc. will be worse than the confidence level values.

Values at 95% confidence level mean that there is only a 5% chance that Net Profit, Drawdown, etc. will be worse than these values.

So Monte Carlo simulation of our strategy shows us that by skipping 10% of random our Net Profit can decrease from \$ 6990 to \$ 3943, and Maximum Drawdown can increase from 6.97% to 11.36%. This means that there is only 5% chance that Net Profit will be lower than \$ 3943. By looking at the higher confidence levels we can see that none of our tests had worse results than \$ 3943, so the strategy seems to be relatively robust to the changes we exposed it to.

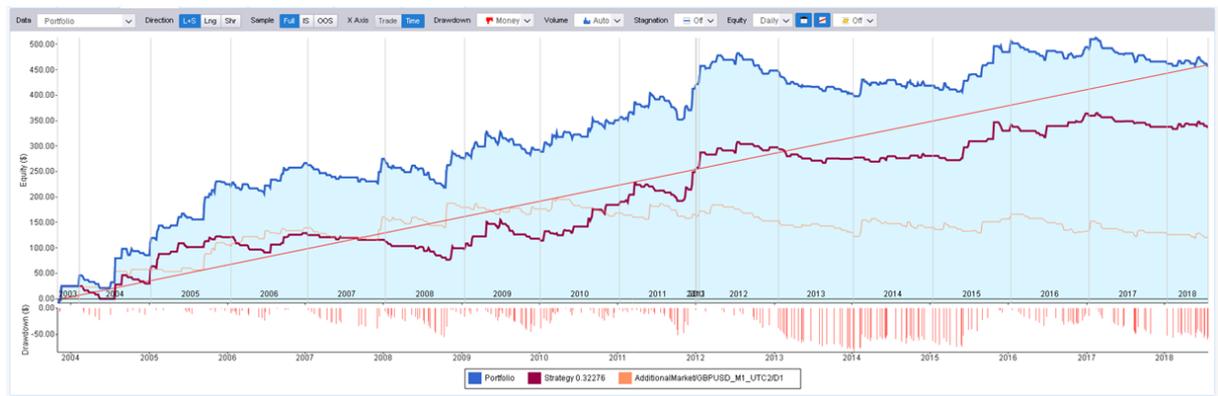
Because Monte Carlo simulations are generated randomly, equity charts and values in the table will slightly differ every time you retest the strategy. Also, the more simulations you'll run, the bigger statistical significance of this test.

5.1.3 Retest on additional markets

This test for robustness is quite tough - it means testing the same strategy on different markets – it means different bol(s) and/or another timeframe(s). Robust strategy should ideally work on multiple symbols/timeframes.

In reality, because each market has its own characteristics, daily volatility, etc., it will be not easy to find a strategy that has the same perfect performance on multiple symbols using just one set of settings.

We can be satisfied if the strategy performs on other markets with at least some degree of profitability, or just slightly losing.



In the two charts above you can see test of strategy on EURUSD (red line), GBPUSD (orange line) and portfolio of both (blue line).

While on the left chart the strategy performs well on both currencies, on the right chart you can see that performance on GBPUSD is bad. This strategy is probably not robust enough.

5.1.4 Monte Carlo retest methods

This is another type of Monte Carlo simulations, in this case it simulates random changes in properties that **require the strategy to be retested** – such as changes in spread, slippage, strategy parameters, or history data.

Because every simulation requires a complete backtest this cross check could take long time. If the backtest on main data took let's say 0.5 seconds, and you want to run 100 simulations in this cross check, you can expect it would take $100 \times 0.5 = 50$ seconds for every strategy where it is applied.

Some of the methods available are:

Randomize Starting Bar - this will test the strategy behavior when the testing starts on a different starting bar. It is obvious that a good strategy cannot be sensitive to which bar you start the test.

Randomize Strategy Parameters - every strategy uses parameters, such as period of an indicator or constant that is used in comparison. This test checks the sensitivity of the strategy to a small change of parameter value. Probability of change is a probability that any parameter changes its value. Max parameter change is the maximum percentage to which the parameter changes its value. For example if you set Max parameter change to 10%, then a parameter with value 60 can be randomly changed to a range 54 - 66 (+- 10% of its original value of 60).

Randomize History Data - one very common case of curve fitting is when strategy is too dependent on history data. This option checks the behavior of the strategy to a change in history data. The Probability of change sets for every bar how probable it is that open, high, low or close price will be changed. The Max price change is a percentage value of the change in relation to ATR (Average True Range).

So if for example close price is randomly chosen to be changed, ATR value is 10 pips, and Max price change is 20%, then the price can change by +- 2 pips

5.1.5 Walk-Forward Optimization

This cross check runs a WF optimization of a strategy. Note that optimization is a slow process.

TO BE ACTUALIZED FOR SQ X

5.1.6 Walk-Forward Matric

The most time demanding type of robustness check is test using Walk-Forward Matrix. If the strategy passes this test it means that with the help of parameter reoptimization it is adaptable to a big range of market conditions.

TO BE ACTUALIZED FOR SQ X

Walk-Forward res...

Overview

List of trades

Equity chart

Trade analysis

Strategy settings

Source code

Multi Walk-Forward Matrix

Total score

Profit

Efficiency

Consistency

Distribution

Drawdown

Trades

Out of sample % / Runs	5	10	15
20	*****	*****	*
30	****	*****	*****

Walk-Forward results for **Out of sample 20% and Runs 10**

Reoptimization period is: reoptimize every 365 days on history of 1458 days

Configure score criteria

[Help](#)

Use the controls below to configure the limit values for each performance property

Total score

5 >= 4

5

Profit

\$5743 >= \$1990

\$ 5743

Efficiency

46% < 60%

46%

Consistency

60% >= 60%

60%

Distribution

41% > 50%

41%

Drawdown

9% <= 25%

9%

Trades

108 > 100

108

5.2 Optimization

5.2.1 Simple Optimization

The idea behind an optimization is simple. First you must have a trading system, this may be a simple moving average crossover for example.

In almost every system there are some parameters (indicator periods, comparative constants, etc.) that decide how given system behave. The optimization means to test the system with different parameter values to find the optimal values of these parameters (giving highest profit or best Return/DD ratio).

Optimization example

Step 1: Loading a strategy for optimization

First, you must switch to Optimizer window and load the strategy you want to optimize.

The screenshot displays the StrategyQuant X Optimizer window. On the left is a sidebar with navigation icons. The main area is split into four sections:

- Optimizer:** Contains 'Stop', 'Pause', and 'Start' buttons. Below are 'Auto scroll' and 'Memory cleanup' checkboxes. It shows 'Total steps: 151' and 'Step: 0'. There are two charts: 'Fitness (in databank)' and 'Heap memory chart'.
- Settings:** Includes 'Data' (Engine: MetaTrader4, Symbol: EURUSD_M1, TimeFrame: H1), 'Optimization type' (Simple selected), and 'Strategy to optimize' (D:\Desktop\SQX manual\EMAcross.sqx). A 'Load' button is highlighted with an orange arrow.
- Results:** Shows 'Best strategy so far: EMAcross' with a table of metrics:

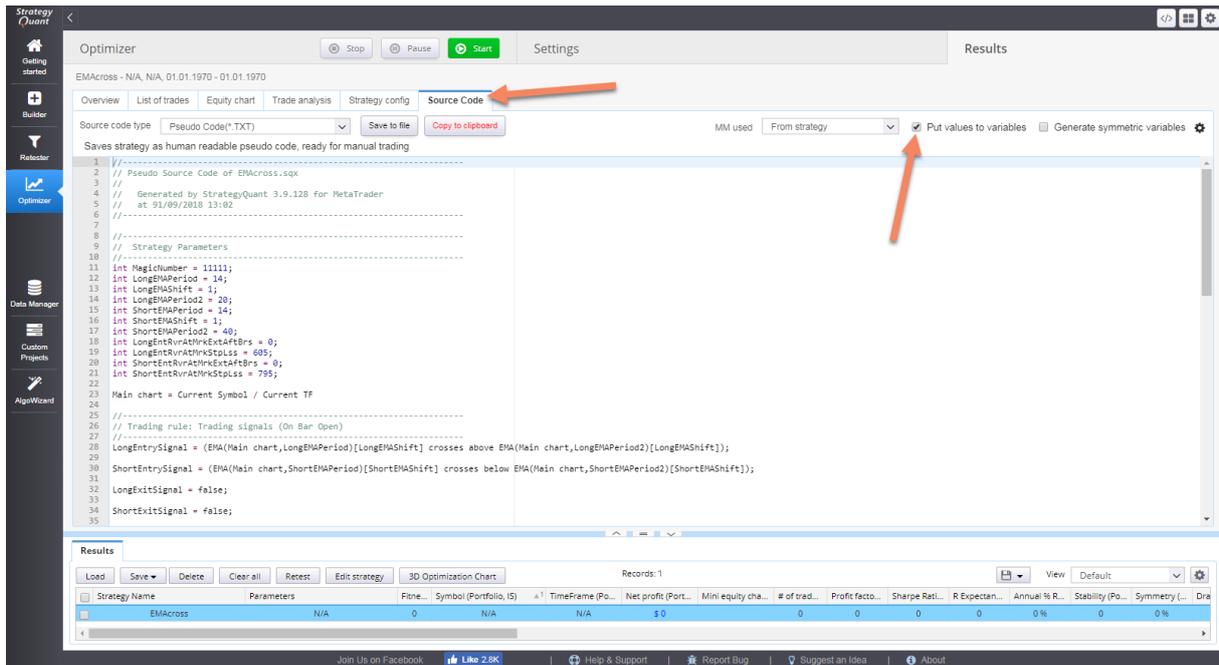
TOTAL PROFIT	\$ -2655.9
YEARLY AVG PROFIT	-4.54 %
YEARLY AVG % RETURN	-4.54 %
CAGR	-5.01 %
# OF TRADES	886
SHARPE RATIO	-0.43
PROFIT FACTOR	0.87
RETURN / DD RATIO	-0.71
WINNING PERCENTAGE	56.6 %
DRAWDOWN	\$ 3737.0
% DRAWDOWN	34.85 %

 Below the table is a line chart showing profit over time, with a shaded 'Out of Sample' region.
- Bottom Results Table:** A table with columns: Strategy Name, Parameters, Fines, Symbol (Portfolio), IS, TimeFrame (Po...), Net profit (Port...), Mini equity cha..., # of trad..., Profit facto..., Sharpe Rati..., R. Expectan..., Annual % R..., Stability (Po...), Symmetry (...), and Dra. The first row shows 'EMAcross' with a net profit of \$ -1936.7 and a Sharpe Ratio of 0.87.

For this example we'll use simple EMA Cross strategy that goes long when faster EMA crosses above slower EMA, and go short when faster EMA crosses below slower EMA.

After you loaded the strategy, it is added also as Original strategy to the Optimization results databank.

You can double click on the Original strategy and then go to Results -> Source code to see its rules.



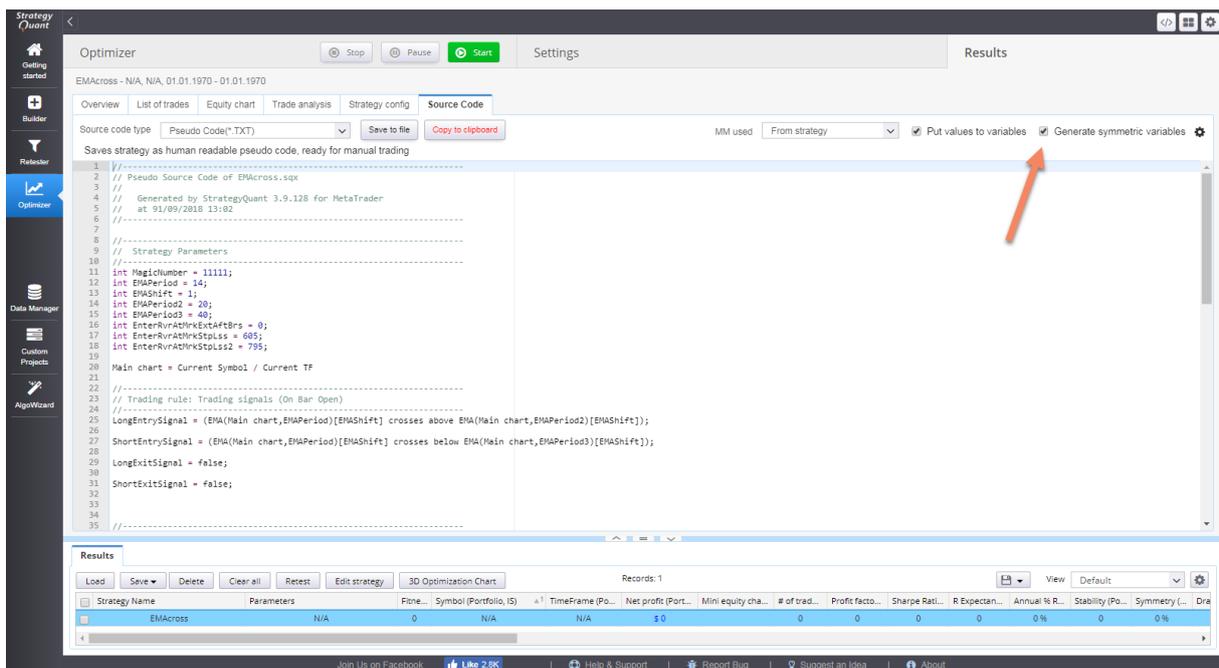
Make sure you check the **Put values to variables** checkbox so that you see that variables LongEMAPeriod, LongEMAPeriod2, ShortEMAPeriod, ShortEMAPeriod2 are used to store indicator parameters.

In our optimization we'll try to find optimal values of these parameters.

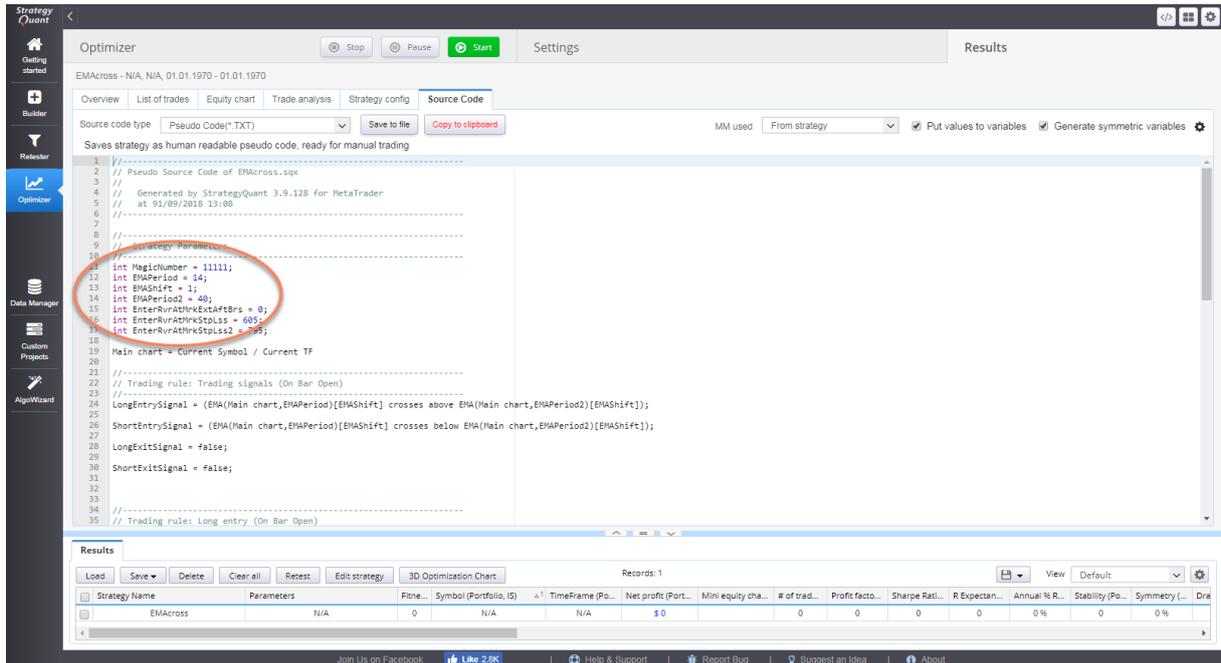
There's still one small problem. We can see that the strategy uses different parameters for long and for short direction.

We can use it like this if we want to find optimal values independently for long and short side, but for our example we'd like to use the same parameter for long and short side.

We can do it using the other checkbox **Generate symmetric variables**



If you check the checkbox, it will use the same parameters for long and short direction (providing the rules are the same).



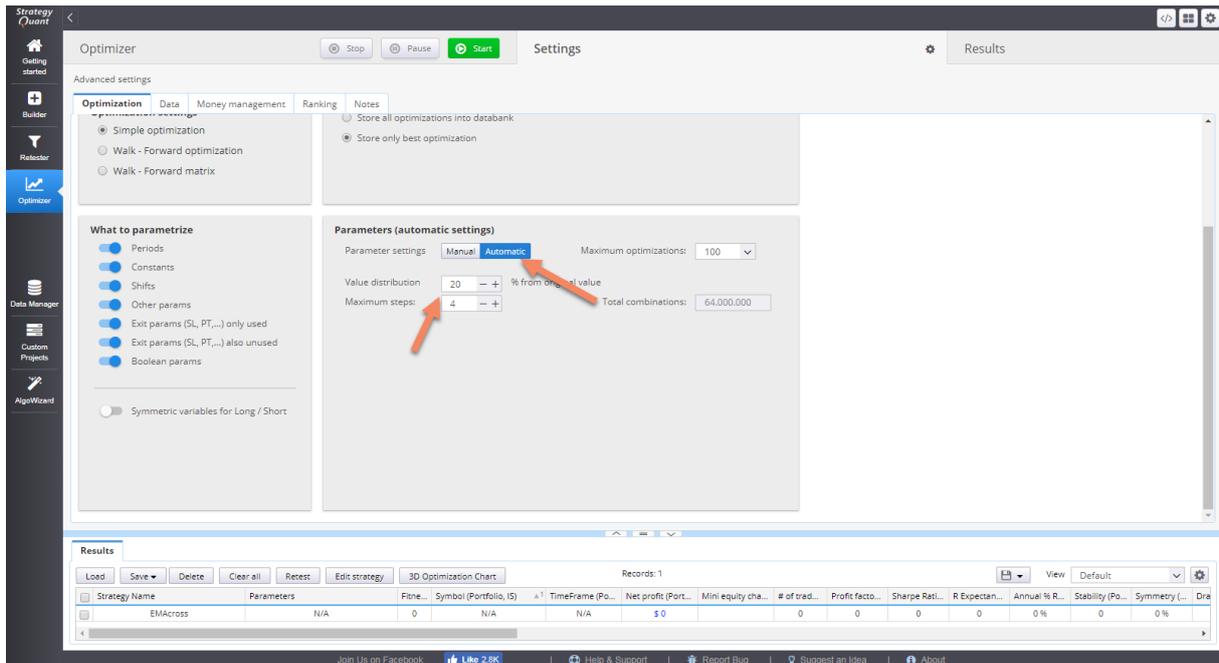
Now you can see that the strategy contains only parameters EMAPeriod, EMAPeriod2 that are used for both directions.

Step 2: Setting optimization values

To set up values that will be optimized we have to go to Settings -> Parameters

Parameters - automatic

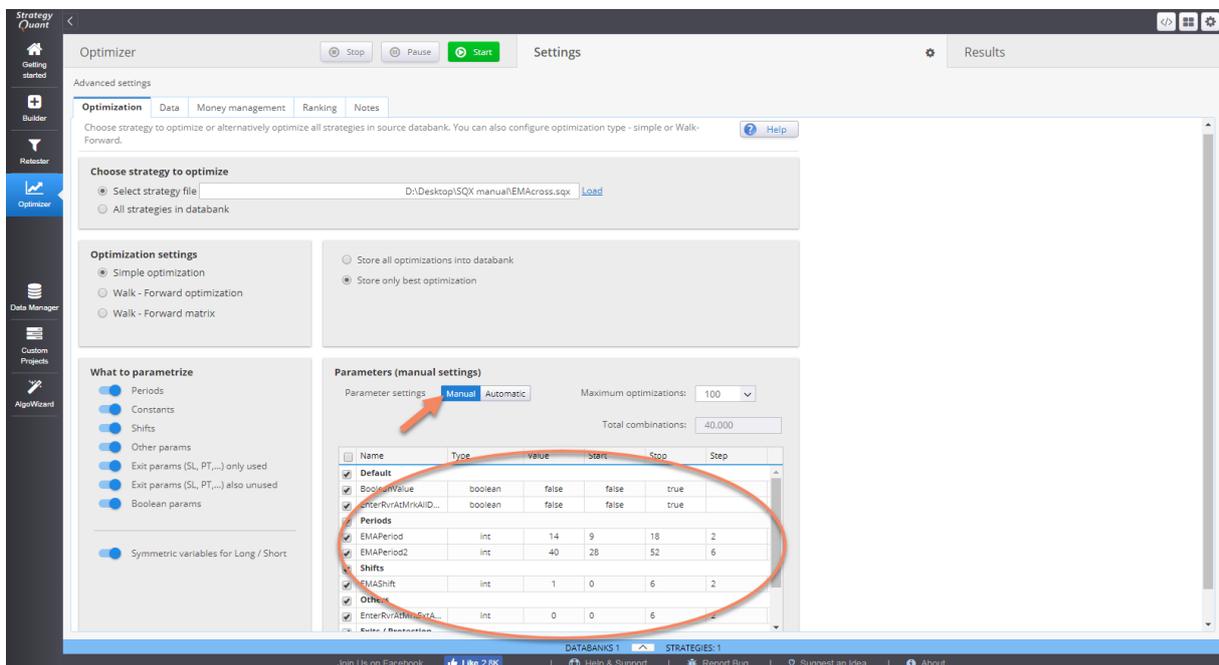
Once this option is selected all parameters will have the value range generated automatically using a distribution process defined by **Value Distribution (%)**. This value defines how much the parameter value will fluctuate by % increments. **Maximum Steps** option defines how many % increments for the parameter value will be generated.



Parameters - manual

Here you can see the list of all strategy parameters that are available for optimization and manual modification.

Optimization simply means trying different values of input parameters.



For every parameter you want to optimize you have to **check** the line of the parameter and choose **Start**, **Stop** and **Step** values. The optimizer will iterate the value from Start to Stop, taking Steps. Original value is also configurable, it will be used to retest the original strategy. You can use this value to compare performance of new results with the "original" settings.

The **Total Combinations** value shows us how many tests have to be performed to test all the combinations of the values. This can be limited using **Maximum Optimizations**.

Note!

It is possible that your parameters table will contain much more parameters, it could look like this:

Parameters (manual settings)

Parameter settings **Manual** Automatic Maximum optimizations: 100 Total combinations: More than 2⁶⁴

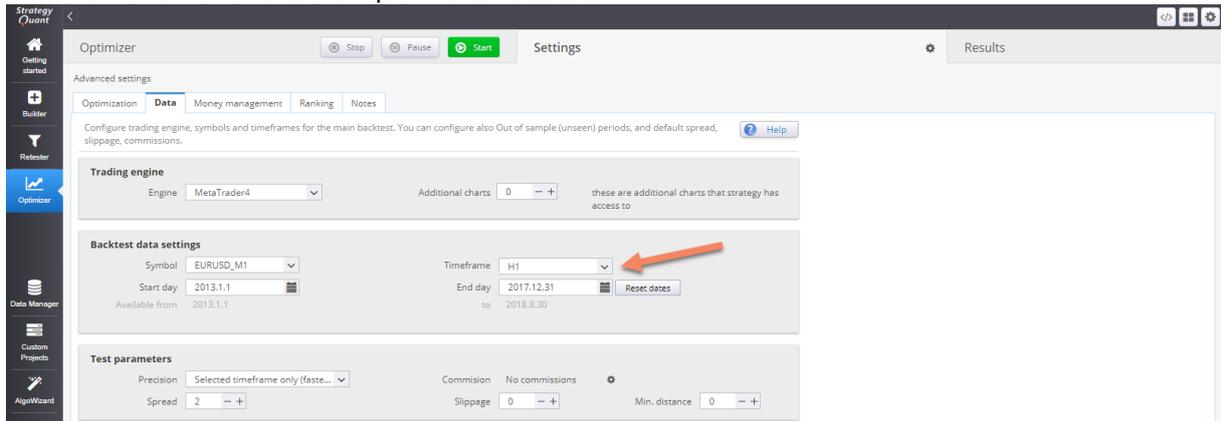
<input type="checkbox"/>	Name	Type	Value	Start	Stop	Step
<input checked="" type="checkbox"/>	ShortATRShift	int	3	2	6	1
<input checked="" type="checkbox"/>	ShortLowShift	int	3	2	6	1
<input checked="" type="checkbox"/>	ShortRSILowerShift	int	1	0	6	2
<input checked="" type="checkbox"/>	Default					
<input checked="" type="checkbox"/>	LongAllowDpiTrd	boolean	false	false	true	
<input checked="" type="checkbox"/>	LongBooleanValue	boolean	false	false	true	
<input checked="" type="checkbox"/>	LongReplaceExisting	boolean	false	false	true	
<input checked="" type="checkbox"/>	ShortAllowDpiTrd	boolean	false	false	true	
<input checked="" type="checkbox"/>	ShortBooleanValue	boolean	false	false	true	
<input checked="" type="checkbox"/>	ShortReplaceExisting	boolean	false	false	true	
<input checked="" type="checkbox"/>	Exits / Protection					

This is another powerful feature of StrategyQuant. It allows you to optimize not only strategy parameters, but also other trading options, such as how many trades to take per day, or what should be the time range for trading or even indicator shifts.

These settings are normally a part of Strategy Options, but you can also optimize their values.

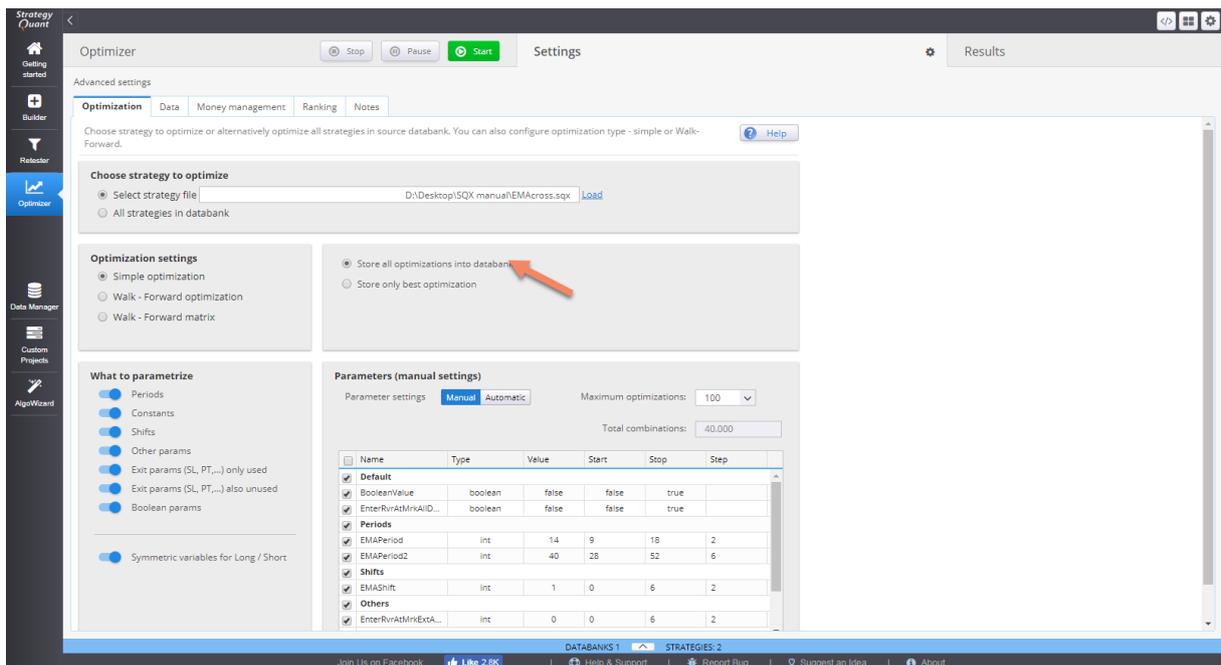
The screenshot shows the StrategyQuant Optimizer interface. On the left, a sidebar contains navigation options: Getting started, Builder, Refresher, Optimizer (highlighted), Data Manager, Custom Projects, and AlgoWizard. The main window is titled 'Optimizer' and includes 'Stop', 'Pause', and 'Start' buttons. Below these are tabs for 'Advanced settings', 'Data', 'Money management', 'Ranking', and 'Notes'. The 'Advanced settings' tab is active, showing options for 'Choose strategy to optimize' (with a file path 'D:\Desktop\ISQX_manual\EMAcross.sqz'), 'Optimization settings' (Simple optimization, Walk-Forward optimization, Walk-Forward matrix), and 'What to parametrize' (Periods, Constants, Shifts, Other params, Exit params, Boolean params, Symmetric variables). A red circle highlights the 'What to parametrize' section. The 'Parameters (manual settings)' section is also visible, showing a table of parameters to be optimized, including 'EMAPeriod', 'EMAShift', and 'EnterRvrAtMinStpLss'.

The last thing we have to configure is the data that will be used for testing. We can choose EURUSD on H1 timeframe for this example in the **Data** tab.



Step 3: Running the optimization

Before we start the optimization process we need to decide how StrategyQuant will store the results. There are two options for this. StrategyQuant can either **Store all optimizations into databank** or **Store only best optimization** which keeps only the best performing modification of the strategy in the databank.



Now we are ready to run the optimization. We'll click on **Start** button.

The optimization engine will test all the possible combinations of selected input parameters and store the results for each combination to the databank on the bottom.

The screenshot shows the StrategyQuant X Optimizer interface. The 'Start' button is highlighted with a red arrow. The interface is divided into several sections: Optimizer, Settings, and Results. The Optimizer section shows a progress bar at 100% and a log of optimization steps. The Settings section shows data for EURUSD_M1 and optimization parameters. The Results section shows a summary of the best strategy found.

The screenshot shows the StrategyQuant X Results interface. The 'Results' table is visible, showing a list of strategies with columns for Strategy Name, Parameters, Fitness, Symbol, TimeFrame, Net profit, Mini equity chart, Profit factor, Sharpe Ratio, R Expectan..., Annual % R..., Stability (Po..., and Symmetry. An arrow points to the 'Net profit' column.

Strategy Name	Parameters	Fitness	Symbol (Portfolio, IS)	TimeFrame (Po...	Net profit (P...	Mini equity cha...	Profit facto...	Sharpe Rat...	R Expectan...	Annual % R...	Stability (Po...	Symmetry
EMACross - Optimization 0.814	BooleanValue=0,EMAPeriod=9,EMAPeriod2=...	0.98	EURUSD_M1	H1	\$ 157070	857	1.24	0.44	0.11	1679.16 %	0	50.27
EMACross - Optimization 0.922	BooleanValue=0,EMAPeriod=9,EMAPeriod2=...	0.98	EURUSD_M1	H1	\$ 159983	857	1.22	0.54	0.1	1018.35 %	0	38.61
EMACross - Optimization 0.617	BooleanValue=0,EMAPeriod=9,EMAPeriod2=...	0.98	EURUSD_M1	H1	\$ 138327	857	1.2	0.46	0.09	4557.71 %	0	32.34
EMACross - Optimization 0.1020	BooleanValue=0,EMAPeriod=9,EMAPeriod2=...	0.98	EURUSD_M1	H1	\$ 131149	857	1.18	0.5	0.08	972.8 %	0	16.55
EMACross - Optimization 0.1019	BooleanValue=1,EMAPeriod=9,EMAPeriod2=...	0.97	EURUSD_M1	H1	\$ 104178	857	1.43	0.02	0.3	37.79 %	0.69	0 %
EMACross - Optimization 0.516	BooleanValue=1,EMAPeriod=9,EMAPeriod2=...	0.97	EURUSD_M1	H1	\$ 94923	857	1.37	-0.41	0.27	-619.39 %	0.34	0 %
EMACross - Optimization 0.415	BooleanValue=1,EMAPeriod=9,EMAPeriod2=...	0.97	EURUSD_M1	H1	\$ 94923	857	1.37	-0.41	0.27	-619.39 %	0.34	0 %
EMACross - Optimization 0.821	BooleanValue=1,EMAPeriod=9,EMAPeriod2=...	0.97	EURUSD_M1	H1	\$ 94923	857	1.37	-0.41	0.27	-619.39 %	0.34	0 %
EMACross - Optimization 0.913	BooleanValue=0,EMAPeriod=9,EMAPeriod2=...	0.97	EURUSD_M1	H1	\$ 88240	830	1.12	-0.41	0.06	-371.84 %	0	0 %
EMACross - Optimization 0.916	BooleanValue=1,EMAPeriod=13,EMAPeriod2=...	0.96	EURUSD_M1	H1	\$ 65238	681	1.33	-0.42	0.24	-1661.32 %	0.32	0 %
EMACross - Optimization 0.618	BooleanValue=1,EMAPeriod=9,EMAPeriod2=...	0.95	EURUSD_M1	H1	\$ 52534	857	1.2	-0.41	0.14	-624.64 %	0.18	0 %
EMACross - Optimization 0.823	BooleanValue=1,EMAPeriod=9,EMAPeriod2=...	0.94	EURUSD_M1	H1	\$ 43562	814	1.12	0.62	0.08	722.11 %	0.3	0 %
EMACross - Optimization 0.321	BooleanValue=1,EMAPeriod=9,EMAPeriod2=...	0.94	EURUSD_M1	H1	\$ 40789	892	1.18	0.82	0.13	43.08 %	0.19	0 %

We can sort the databank by Net Profit and we can see that the best input values in terms of maximum profit are EMAPeriod = 9 and EMAPeriod2 = 40.

Note!

You can see certain results having an exclamation mark shown next to the strategy title. This means for given parameters set the optimizer found the strategy behaving oddly and it needs to be reviewed manually in more detail.

Interpreting the results

Now we've got input parameters that were optimized for our given symbol and timeframe.

What we've really done is we found out what worked best in the past. We have to be very careful because the parameters might be ideal for the history data, but there is no guarantee that what worked best on history data will work also in the future.

It is called curve fitting - usually the more parameters the strategy has, the bigger is the danger of curve fitting.

There are two approaches to curve fitting:

- make sure the strategy is robust and not optimizing its values at all
- make sure the strategy benefits from periodic reoptimization

So the question is - will periodical reoptimization improve the results of my strategy?
If yes, how often should I do it?

StrategyQuant can give you answers to these questions using another of its advanced functionality - Walk-Forward Optimization and Walk-Forward Matrix.

5.2.2 Walk-Forward Optimization

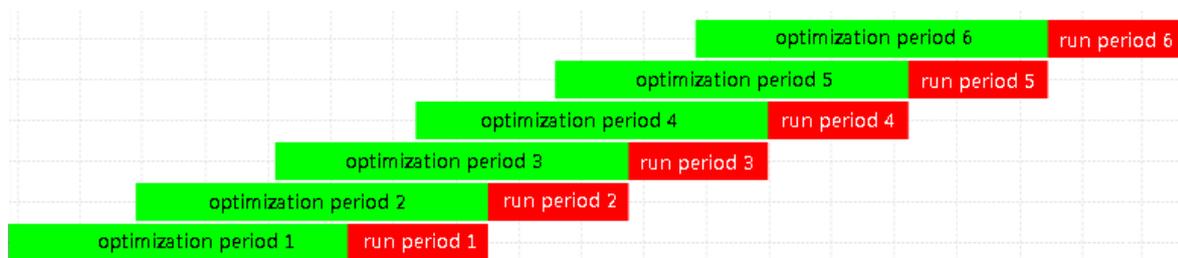
Walk-Forward optimization is generally a set of simple optimizations followed by trading using the best parameters.

It is a technique in which you optimize the parameter values on a past segment of market data, then verify the performance of the system by testing it forward in time on data following the optimization segment, and the process can be repeated over subsequent time segments.

How Walk-Forward optimization works

in walk-forward optimization, the data are divided into a configurable number of periods (6 in this example).

Each period consists of optimization part and run part.



The program starts with optimization period 1. It will run the simple optimization on optimization period 1 to find the best parameter values.

These parameter values are then applied to run period 1 - strategy is tested by trading with the optimized parameters on the run period.

At the end of run period 1, the system again runs simple optimization on a part of data marked as optimization period 2.

It finds the best set of parameter values and they are again used for trading in run period 2.

This continues until the period 6, which is also the end of history data used in test.

Walk-Forward optimization simulates how you'd work with the strategy during real trading - you'd optimize it on some historical data and then trade it with the optimal values.

After some time you'd want to reoptimize it and let it trade again.

Analysis of walk-forward results can tell you if the strategy is a good candidate for periodic reoptimization.

Walk-Forward optimization example

Please refer to an article from our website for the Walk-Forward optimization example:

<http://www.strategyquant.com/articles/walk%20forward%20optimization>

5.2.3 Walk-Forward Matrix

Walk-Forward Matrix is a powerful feature that can help you with two things:

1. Find the optimal period for strategy reoptimization

it will help you identify the best optimization frequency

2. Test the strategy for robustness

if the strategy passes Walk-Forward Matrix test it means that with the help of parameter reoptimization it is adaptable to a big range of market conditions.

Standard Walk-Forward optimization tests the strategy results if it is periodically reoptimized, let's say every 300 days.

But how do we know how often it is the best to reoptimize the strategy?

We can only guess, unless we'll use Walk-Forward Matrix that will test various combinations of reoptimization periods.

Walk-Forward Matrix example

Please refer to an article from our website for the Walk-Forward optimization example:

<http://www.strategyquant.com/articles/walk%20forward%20matrix>

5.3 Portfolios

There are two possibilities how to create a portfolio:

1. Use [Additional data](#) to build/test strategy on multiple symbols or timeframes - this will create a portfolio that will contain results of the same strategy tested on different data
2. Manually [combine strategies to a portfolio](#) - this will create a portfolio of different strategies combined together, trading on the same or different symbols & timeframes

Portfolio results are a summary results of all the strategies it contains. For portfolios created manually (2) there is a special "Portfolio #" result created in databank that can be saved to a file for later use.

Portfolio chart

Equity chart for portfolio combines equity of the portfolio as well as equity for every strategy in the portfolio. When viewing a portfolio you can switch between portfolio chart and charts of substrategies using a combo box on the top left corner of the chart.



5.4 Controlling conditions negation

When you are generating symmetric strategies, you usually want to have short rule opposite (a negation) of a long rule.

For example, if you have:

Long rule: CCI(14) crosses above 0

then you want your short rule to be a negation of that:

Long rule: CCI(14) crosses below 0

This is a standard behaviour - normally an indicator tells us buy or sell, but sometimes a building block / indicator is a **trade or no trade decision**. The most common type is a volatility indicator, or some external indicator data.

For example:

1. Volume falls to low level
2. ATR falls to a low level
3. StdDev of price falls to a low level
4. External volatility such as VIX

The key is the signal is the same for long and short.

In StrategyQuant every block has its opposite block defined, so the program knows how to negate the block from long rule for short rule, and for most users it works by default and doesn't need touching.

For some advanced users who want more control, there is a way to override this functionality.

In {StrategyQuant}\user\settings folder there is a file **OppositeBlocks.csv**

As its name suggests, it contains a definition of opposite block for every existing block. Its format is simple:

```
Block;OppositeBlock
ADXChangesDown;ADXChangesUp
ADXChangesUp;ADXChangesDown
```

Where ADXChangesDown and ADXChangesUp and constants or codes of blocks "ADX changes direction downwards" and "ADX changes direction upwards".

It says that when *ADXChangesDown* is used in long rule, then we should use its opposite block *ADXChangesUp* in the short rule, and vice versa.

By changing the opposite block in this configuration file you'll tell SQ X to use another block – or use the same block as opposite. You have to restart SQ before this change will take effect.

So if you'll change the line:

ADXChangesDown;ADXChangesUp

to

ADXChangesDown;ADXChangesDown

it will generate strategies where in both long and short rules there will be the same condition: "ADX changes direction downwards"

There are some restrictions – only block with the same number and names of parameters can be used as opposite block.

You cannot define for example MACD as an opposite block of CCI, because each of these indicators has different set of parameters.

Another restriction is in comparisons. SQ X uses signals heavily – signals are predefined conditions containing a combination of blocks, such as "ADX crosses above Level".

The same condition could be created also as a connection of three independent blocks:

[ADX] [crosses above] [Number].

but doing it like this has two major disadvantages:

- Remember that each of these three parts is generated totally randomly, there is a small chance that exactly this condition will ever be generated
- How would you control how it negates to opposite block when it is in fact three separate blocks?

For this reason, SQ X uses primarily signal blocks, which are predefined most often used comparisons.

StrategyQuant allows you to create your own indicators and signals in Code Editor, so there is no limit to custom signals you can create.

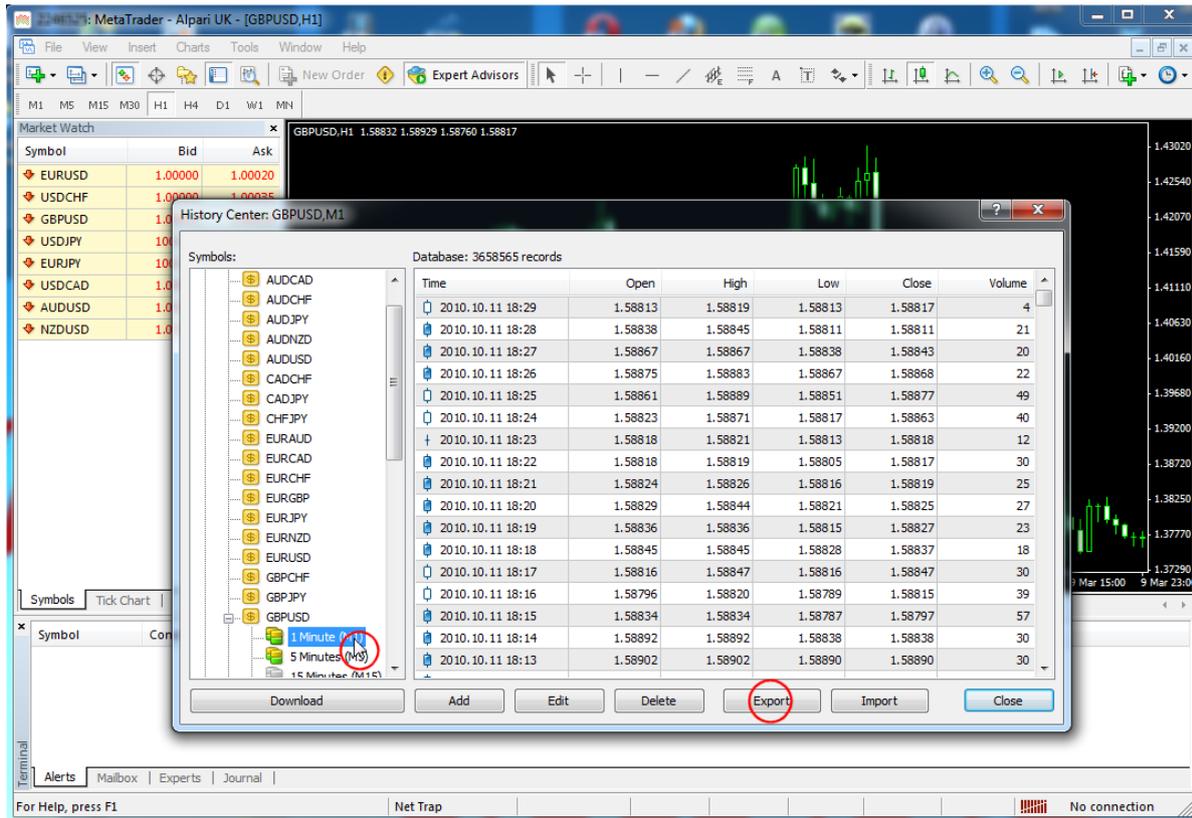
6 How to...

6.1 Import history data from MetaTrader 4

TO BE ACTUALIZED FOR SQ X

Step 1: Export data from Metatrader

Open your Metatrader and go to **Tools -> History Center**.



There, open the currency you want to export (for example GBPUSD) and double click on **1 Minute (M1)** so that it is refreshed on the right side of the screen.

Then just click on the **Export** button and choose your destination file.

Note

StrategyQuant supports import of only 1 Minute data, it will compute the higher timeframes automatically.

Now we have the data ready to be imported to StrategyQuant.

Step 2: Create a new symbol in StrategyQuant

We can import data to already existing symbol, but that will overwrite its data, so it is better to define a new symbol.

Start StrategyQuant, go to **Data manager** screen, **History Data** and click on **Add symbol...**

The screenshot shows a dialog box titled "Add new symbol" with a close button (X) in the top right corner. The dialog contains a "Symbol Info" section with the following fields and values:

- Symbol: GBPUSD_mt4
- Symbol Full Name: GBPUSD_mt4
- Pip/Tick Point: 0.0001 (with a dropdown arrow) and the text "what is one pip unit?" below it.
- Pip/Tick Value in \$: 10 (with a dropdown arrow) and the text "what is the money value of one pip?" below it.
- Pip/Tick Step: 0.00001 (with a dropdown arrow) and the text "what is minimum pip step?" below it.
- Default Spread: 3 (with a dropdown arrow) pips
- Cost per roundturn: 0 (with a dropdown arrow)

At the bottom right of the dialog are "OK" and "Cancel" buttons.

Fill out the symbol and full name, check the default spread and tick value.

Pip/Tick point

is the value of pip. It means what number is 1 pip. is usually 0.0001. For **JPY** based pairs 1 pip is 0.01.

Pip/Tick value in \$

is a value how much is one pip worth in money, it is usually 10 for all the currencies.

Pip/Tick Step

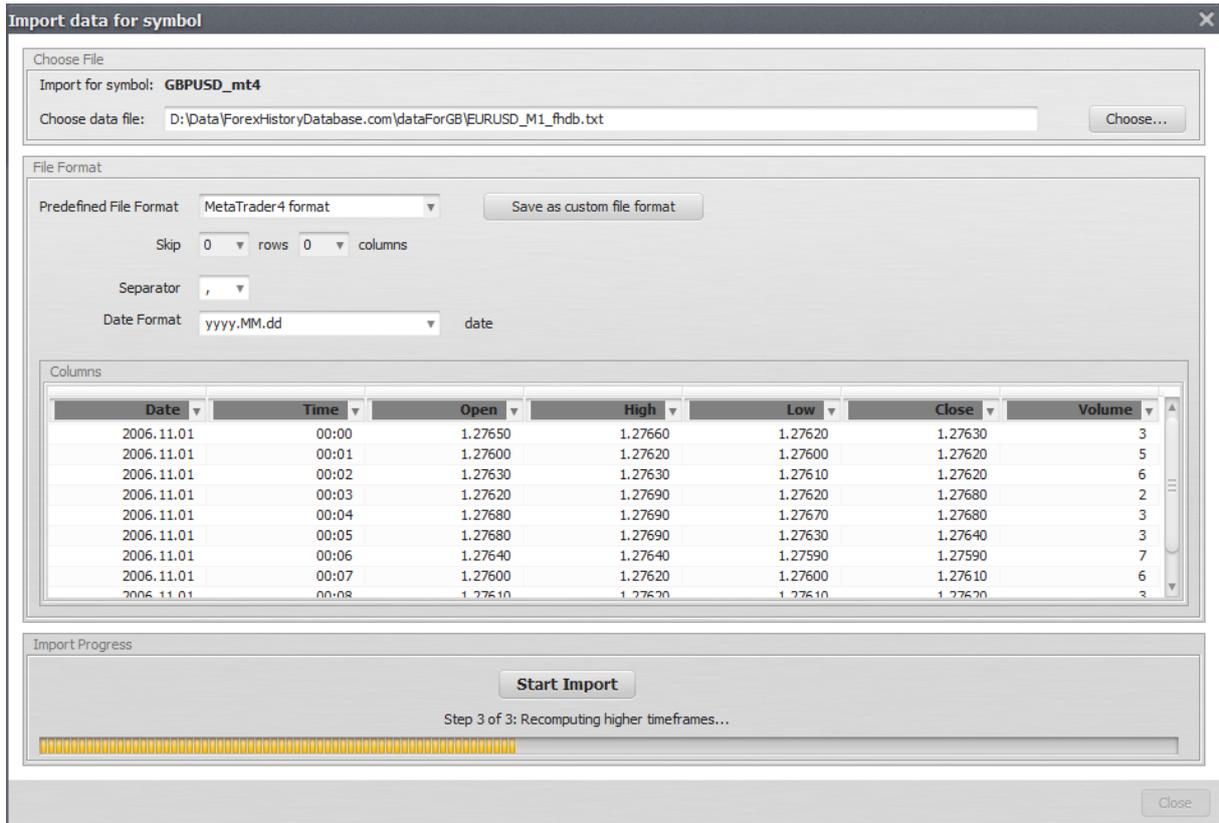
is a value by how much one pip can move. Virtually all brokers now use 5-digit data, so the value will be 0.00001 (or 0.001 for JPY based pairs).

Click OK and the new symbol will be created. The symbol doesn't have any data yet, but we are going to import them in the next step.

Step 3: Import data to StrategyQuant

Now select your new row with the symbol and click on **Import data ...** button on the top. This will open the import dialog.

Import dialog is configurable; it allows importing data from various file formats. Since we use data from MetaTrader, choose MetaTrader4 as a Predefined File Format.



Choose the data file and click on Start Import button. This will start the import process. It could take few minutes, depending on speed of your computer and data size.

When the import is finished it will display information window and asks you to close the dialog.

Now we have the new data successfully imported into StrategyQuant and we can use them for tests or new strategies generation.

6.2 Export strategy from StrategyQuant and test or trade it in MetaTrader

TO BE ACTUALIZED FOR SQ X

When you generate some strategies and find the ones you would potentially like to use in real trading, it is time to test them in MetaTrader.

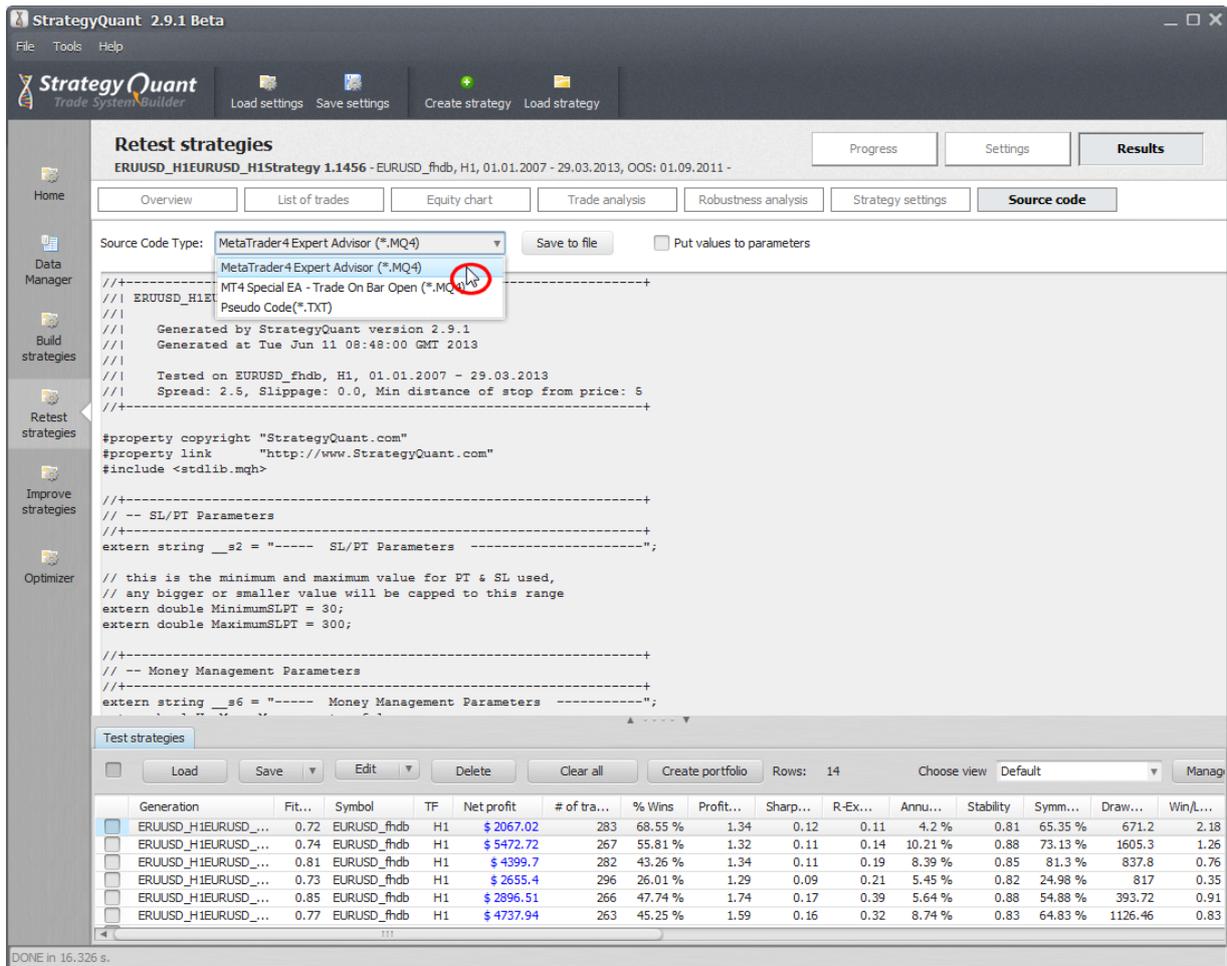
StrategyQuant normally saves strategies in its own proprietary .str file format, which is not readable by MetaTrader.

In order to test strategies in MT4 you have to export its source code in MQL format

This is simple, go to the databank and find the strategy you want to use. Double-click on it, which opens it in the Result details window above the Databank.

Generation	Fit...	Symbol	TF	Net profit	# of tra...	% Wins	Profit...	Sharp...	R-Ex...	Annu...	Stability	Symm...	Draw...	Win/L...
ERUUSD_H1EURUSD_...	0.72	EURUSD_ffdb	H1	\$ 2067.02	283	68.55 %	1.34	0.12	0.11	4.2 %	0.81	65.35 %	671.2	2.18
ERUUSD_H1EURUSD_...	0.74	EURUSD_ffdb	H1	\$ 5472.72	267	55.81 %	1.32	0.11	0.14	10.21 %	0.88	73.13 %	1605.3	1.26
ERUUSD_H1EURUSD_...	0.81	EURUSD_ffdb	H1	\$ 4399.7	282	43.26 %	1.34	0.11	0.19	8.39 %	0.85	81.3 %	837.8	0.76
ERUUSD_H1EURUSD_...	0.73	EURUSD_ffdb	H1	\$ 2655.4	296	26.01 %	1.29	0.09	0.21	5.45 %	0.82	24.98 %	817	0.35
ERUUSD_H1EURUSD_...	0.85	EURUSD_ffdb	H1	\$ 2896.51	266	47.74 %	1.74	0.17	0.39	5.64 %	0.88	54.88 %	393.72	0.91
ERUUSD_H1EURUSD_...	0.77	EURUSD_ffdb	H1	\$ 4737.94	263	45.25 %	1.59	0.16	0.32	8.74 %	0.83	64.83 %	1126.46	0.83

There, go to Source code tab and switch the source code to MetaTrader4 Expert Advisor. This will load the MT4 code of the strategy.



Click on Save to file button, and in the file dialog find the folder where your MetaTrader is installed (for example C:\Program Files\Alpari MT).

In this folder you have to go inside the **experts** directory and save the strategy source code there.

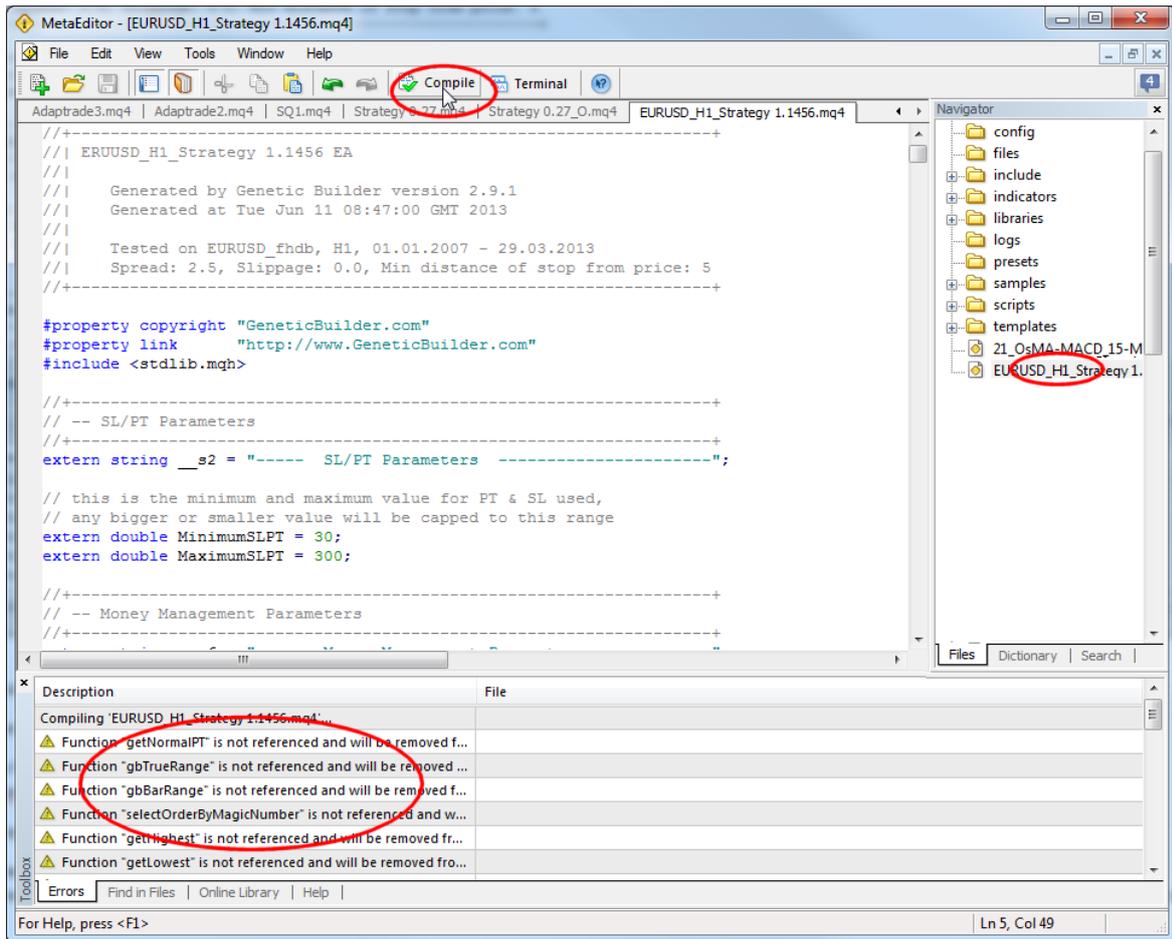
So the full path of the file will be for example

C:\Program Files\Alpari MT\experts\ERUUSD_H1_Strategy 1.1456.mq4

Now the strategy is copied to Metatrader. You can open MetaTrader now.

In Metatrader go to menu **Tools -> MetaQuotes Language Editor**, or press **F4**. This will open the language editor.

On the right side of the editor you'll have a list of strategies that are in the **experts** folder. Double-click on our strategy to open it in the editor window and then click on **Compile** on the top toolbar.



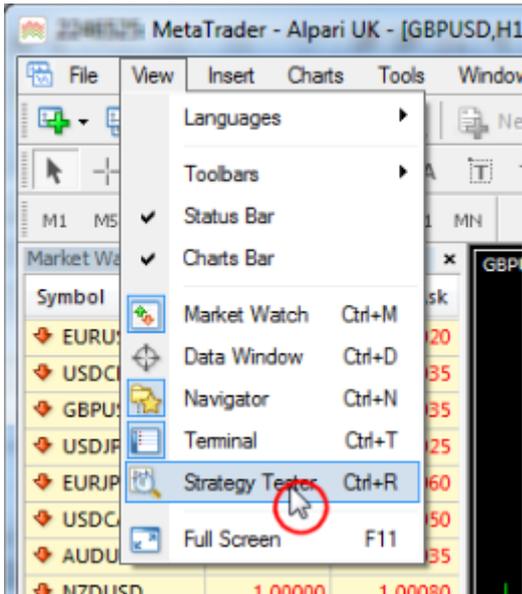
The strategy will be compiled and now it is ready for backtest or running live.

Note - Compilation warnings are normal

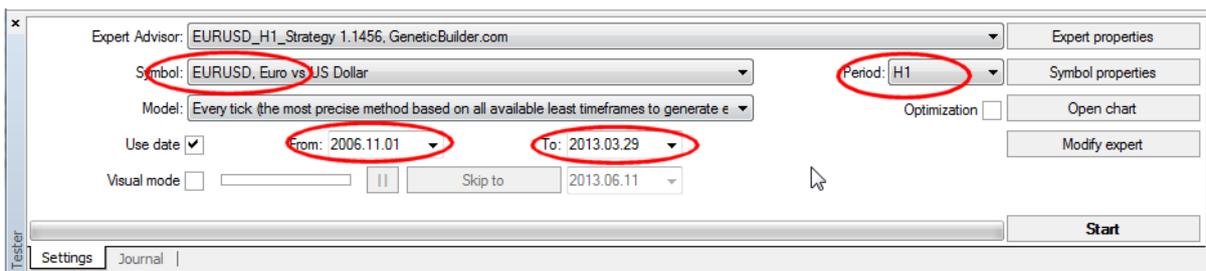
Please note that there are some compilation warnings on the bottom. These warnings are normal and they don't influence the strategy work.

There are simply some functions that are not used in the strategy and MetaTrader is informing you about that.

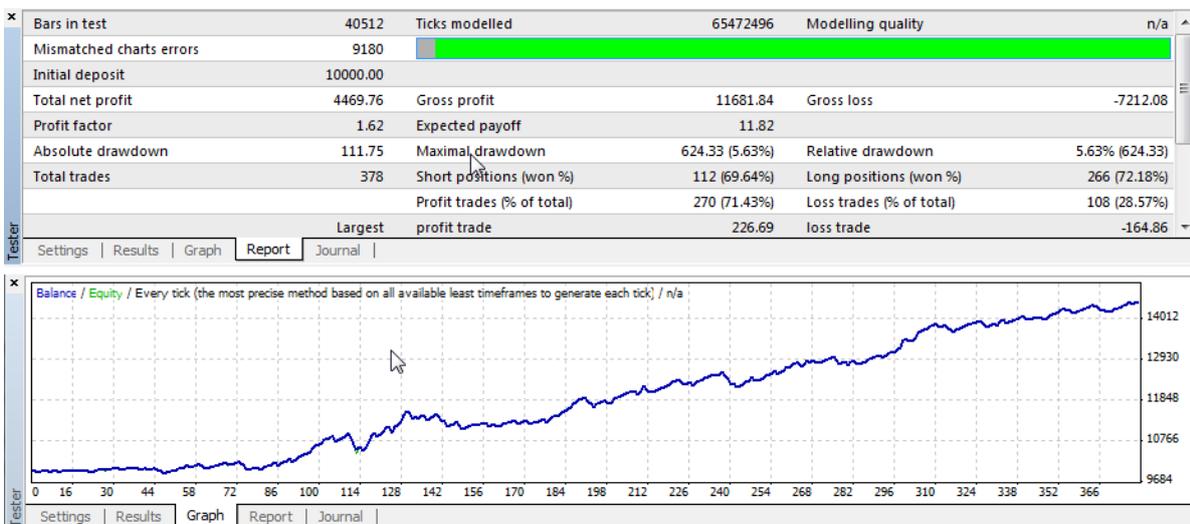
Now that the strategy is compiled , it is ready to be backtested. You can close the **MetaEditor**, go to the main MetaTrader screen and open **Strategy Tester**.



This will open the **Strategy Tester** dialog on the bottom and you can run the backtest.



Make sure you select the correct Expert Advisor, Symbol, Timeframe and Date From and To and then click on the **Start** button. The test will start and after a while you'll get the results:



Explanation of small differences in backtests

If you'll compare test results in StrategyQuant and in MetaTrader, you'll see that on some cases the

backtesting results are not the same.

The results can differ slightly or significantly - depending on the type of strategy.

Backtesting algorithm used in StrategyQuant is very accurate, but it is not exactly the same algorithm used in MetaTrader, so it produces slightly different result.

The important thing here is to understand that both testing algorithms are only approximations, one isn't superior to the other.

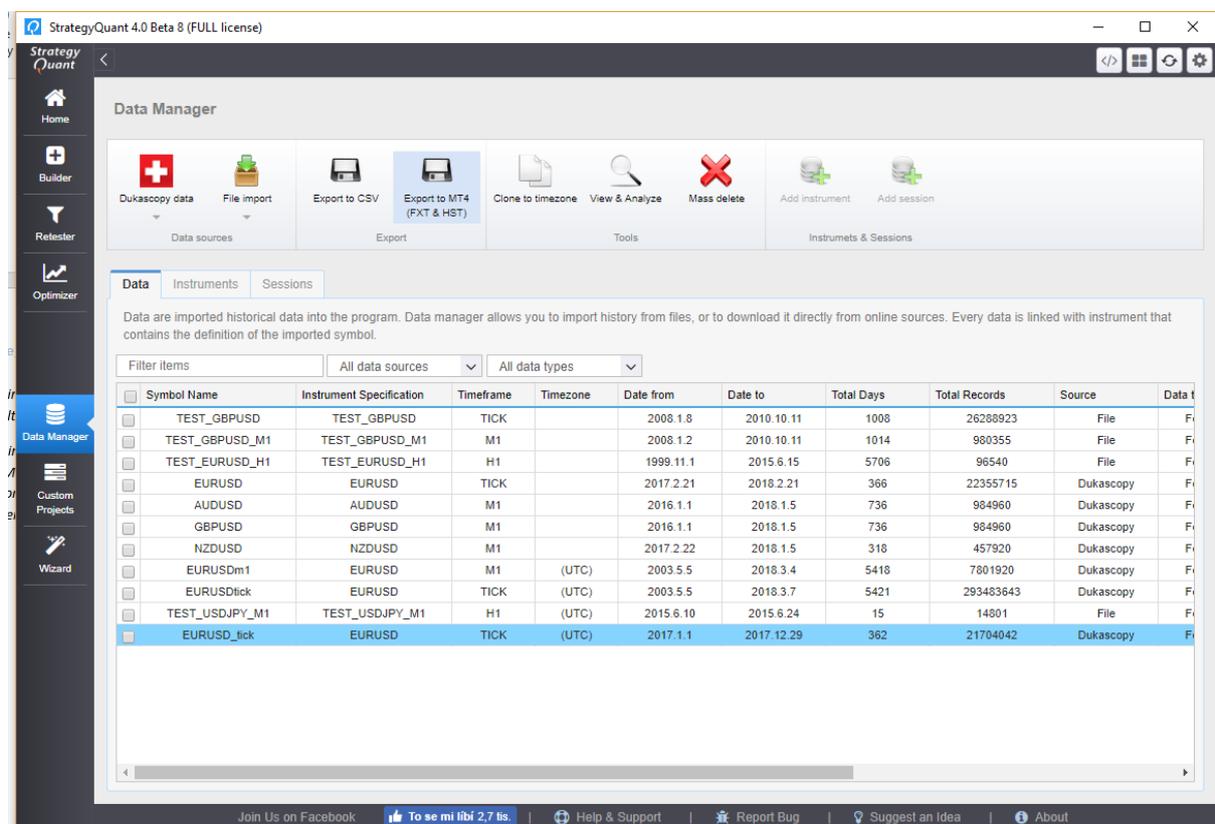
6.3 Test strategy in MetaTrader 4 with tick precision

New StrategyQuant4 Data Manager now allows you to export special FXT & HST files for MetaTrader 4 that allow you to test your strategies in MT4 with the highest possible modeling quality.

It does it by exporting tick data from StrategyQuant and using them in MT4 Strategy Tester.

Step 1: Export FXT & HST data to MetaTrader4

Go to Data manager, choose some tick data and click on Export to MT4 (FXT & HST) button on the toolbar.



This will open a new dialog where you must specify more details.

First of all you should specify the date range to be exported.

If you are doing it for the very first time, you also must specify the path to your MetaTrader4 installation. SQ X needs it so that it generates the files in the appropriate folders.

The last thing is to choose the correct MT4 symbol and its setting on the lower half of the dialog. The symbol is usually recognized automatically, but if you use some exotic forex symbol you might need to choose the symbol yourself and even perhaps export the actual settings of your MetaTrader. This is explained more in the Troubleshooting section.

Export to MetaTrader 4 FXT & HST
✕

Choose data range to export

From

To

Please note that MT4 has file-size limit of maximum 4GB, it won't be able to process bigger data files.

Output folder

MT4 Installation [Select](#)

MT4 Data Folder

Server name ▼

Symbol ▼ Name in MT4

[Load other MT4 data specification file](#)

Margin stopout 0 Margin mode 0 Mode marginrequired 0 Freeze level 0 Profit mode 0 Swap short 0 Tick size 1e-05 Lot step 0 Margin maintenance 100000 Margin initial 100000 Contract size 100000 Tick value 0	Mode point 0 Margin stopout_mode 0 Margin hedged 50000 Free margin_mode 0 Stops level 0 Lot max 0 Spread 3 Swap type 0 Mode bid 1 Swap long 0 Lot min 0 Mode margincalcmode 0
--	--

[Close](#)

When everything is configured properly click on Start export. Data Manager will now start generation of FXT & HST files that will be used in MT4 Strategy Tester to achieve high quality backtest.

Note! You must turn off MetaTrader before export, otherwise there might be errors in the exported files.

Step 2: Start MetaTrader and run strategy backtest

Start your MetaTrader, open Strategy Tester, choose some strategy to be tested and the symbol you just exported. You can leave the Use date unchecked – it will perform test on the whole exported data.

When your backtest finishes go to Report and you should see that the backtest was made with 99% modeling quality.

6.3.1 Possible issue – Custom MT4 settings or missing symbol

When you use some exotic symbol or you want to use exactly the same trading settings as in your MT4 installation you can use our script to export your MT4 symbols details.

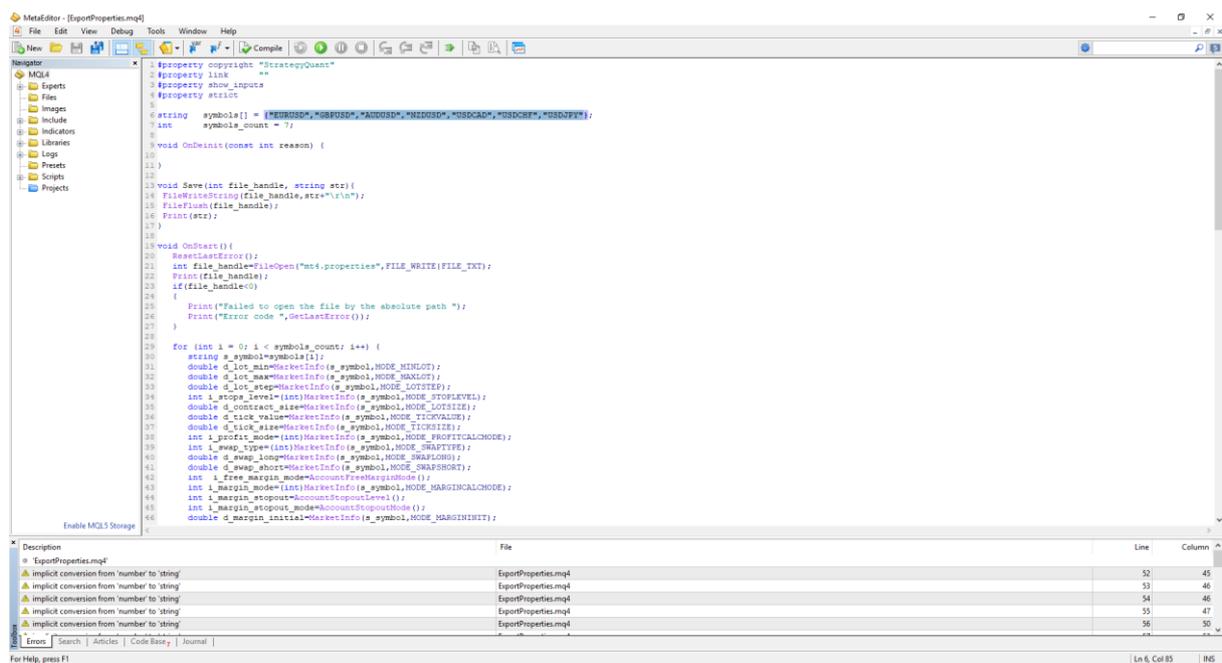
To do that you have to run our specila script that exports MT4 symbol detalt to a file for use in Data manager.

The custom script is in folder **{SQ installation}/custom_indicators/MetaTrader4/Scripts**, it is named **ExportProperties.mq4**

Copy this script to your **{MT4 Data folder}/MQL4/Scripts** folder and restart MetaTrader.

Then go to Tools -> MetaQuotes Language Editor and open this script. By default it exports properties for the symbols: EURUSD,GBPUSD,AUDUSD,NZDUSD,USDCAD,USDCHF,USDJPY specified in the array in the beginning of the script. If yout broker uses different symbol names or you want to test another symbols add them to this array.

Then click on Compile to compile the updated script.



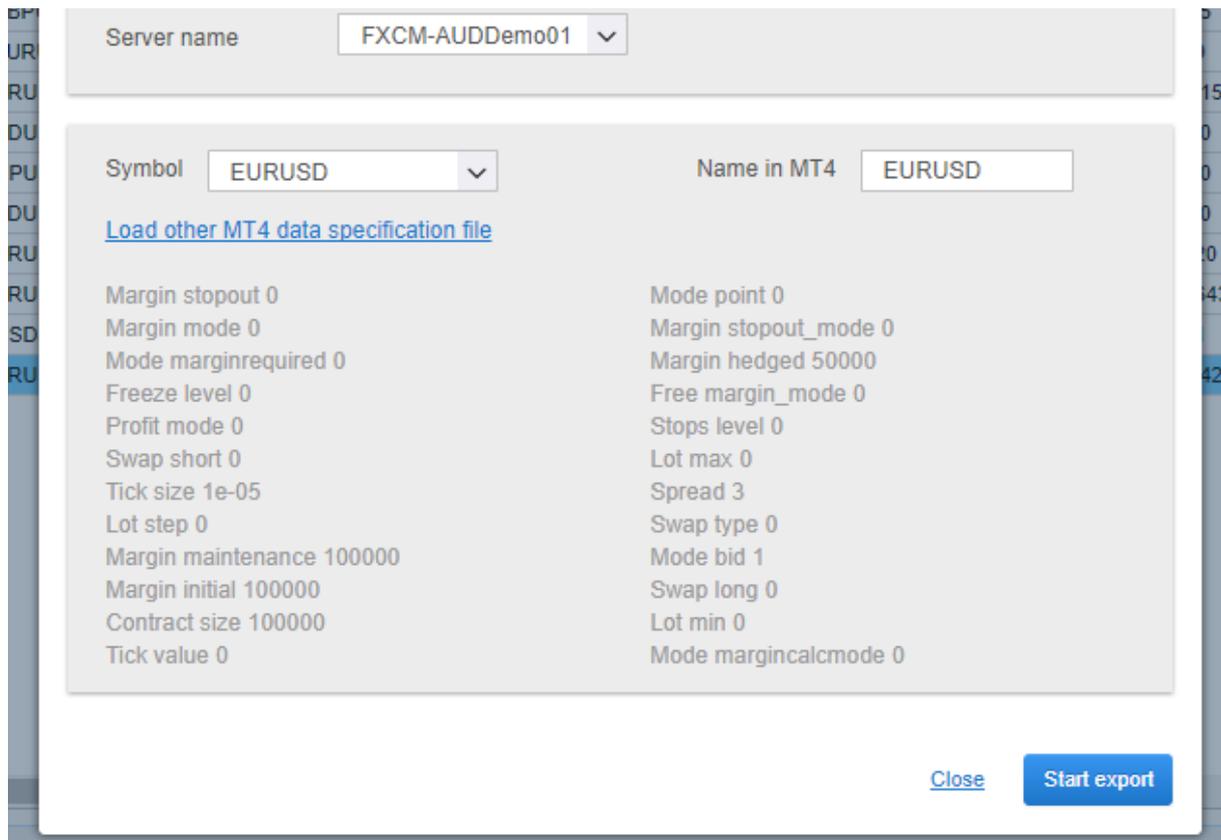
Now you must start the script on any open chart. You can do this by going back to MetaTrader, finding the script in the Navigator and drag & drop it to some chart.

Make sure that your MetaTrader is connected to the broker, otherwise it wouldn't work.

When the script finishes it will export the properties of configured symbols to a file **{MT4 Data folder}/MQL4/Files/mt4.properties**

Copy this file to some folder where you can easily find it later.

Now in the Step 1 where you configure your export click on the link **Load other MT4 data specification file** and find your file.



It will load your newly generated file with the current properties for every symbol in your MetaTrader. Then just chose the right symbol, click Export and it is done.

6.3.2 Possible issue – Backtest doesn't produce trades or there are errors in the Journal

Backtest doesn't produce trades or there are many errors in the Journal.

Time	Message
2018.04.09 10:44:59.633	2017.12.26 21:59:45 MACD Sample EURUSD,H1: OrderSend error 131
2018.04.09 10:44:59.633	2017.12.26 21:59:44 MACD Sample EURUSD,H1: Error opening SELL order : 131
2018.04.09 10:44:59.633	2017.12.26 21:59:44 MACD Sample EURUSD,H1: OrderSend error 131
2018.04.09 10:44:59.633	2017.12.26 21:59:44 MACD Sample EURUSD,H1: Error opening SELL order : 131
2018.04.09 10:44:59.633	2017.12.26 21:59:44 MACD Sample EURUSD,H1: OrderSend error 131
2018.04.09 10:44:59.633	2017.12.26 21:59:44 MACD Sample EURUSD,H1: Error opening SELL order : 131
2018.04.09 10:44:59.633	2017.12.26 21:59:44 MACD Sample EURUSD,H1: OrderSend error 131
2018.04.09 10:44:59.633	2017.12.26 21:59:44 MACD Sample EURUSD,H1: Error opening SELL order : 131
2018.04.09 10:44:59.633	2017.12.26 21:59:44 MACD Sample EURUSD,H1: OrderSend error 131
2018.04.09 10:44:59.633	2017.12.26 21:59:44 MACD Sample EURUSD,H1: Error opening SELL order : 131

Most probable reason is that the properties of the symbol used during FXT export don't match settings of your broker. You have to export your own symbol properties from MT4 and use them in Data manager using the steps described in **Possible problem 1**.

6.3.3 Possible issue – Modelling quality is n/a

Sometimes the modeling quality in MT4 results page isn't shown as 99%, but as **n/a**.

Bars in test	7041	Ticks modelled	25522812	Modelling quality	n/a
Mismatched charts errors	0				
Initial deposit	10000.00			Spread	Current (19)
Total net profit	123.80	Gross profit	4789.70	Gross loss	-4665.90
Profit factor	1.03	Expected payoff	0.45		
Absolute drawdown	497.60	Maximal drawdown	694.70 (6.81%)	Relative drawdown	6.81% (694.70)
Total trades	276	Short positions (won %)	138 (28.26%)	Long positions (won %)	138 (34.78%)
		Profit trades (% of total)	87 (31.52%)	Loss trades (% of total)	189 (68.48%)
	Largest	profit trade	244.80	loss trade	-180.60
	Average	profit trade	55.05	loss trade	-24.69
	Maximum	consecutive wins (profit in ...)	5 (229.70)	consecutive losses (loss in ...)	10 (-194.50)
	Maximal	consecutive profit (count of...)	244.80 (1)	consecutive loss (count of l...)	-348.70 (9)
	Average	consecutive wins	2	consecutive losses	3

This is less serious than it seems, it means that MetaTrader is unable to determine the modeling quality. We haven't found a reason why it sometimes happens, it seems it is related to particular MT4 installation.

However, as we see in the picture above, **despite modelling quality being n/a there are no errors, and the whole modeling bar is green.**

So it means the backtesting worked on real tick data with no errors, it is a problem of MetaTrader that it cannot compute the correct modelling quality for some reason.

It is usually an issue in the particular MetaTrader installation, another installation of MetaTrader running on the same FXT data will return 99% modeling quality.

Bars in test	7041	Ticks modelled	25522812	Modelling quality	99.00%
Mismatched charts errors	0				
Initial deposit	10000.00			Spread	Current (19)
Total net profit	123.80	Gross profit	4789.70	Gross loss	-4665.90
Profit factor	1.03	Expected payoff	0.45		
Absolute drawdown	497.60	Maximal drawdown	694.70 (6.81%)	Relative drawdown	6.81% (694.70)
Total trades	276	Short positions (won %)	138 (28.26%)	Long positions (won %)	138 (34.78%)
		Profit trades (% of total)	87 (31.52%)	Loss trades (% of total)	189 (68.48%)
	Largest	profit trade	244.80	loss trade	-180.60
	Average	profit trade	55.05	loss trade	-24.69
	Maximum	consecutive wins (profit in money)	5 (229.70)	consecutive losses (loss in money)	10 (-194.50)
	Maximal	consecutive profit (count of wins)	244.80 (1)	consecutive loss (count of losses)	-348.70 (9)
	Average	consecutive wins	2	consecutive losses	3

Trading orders page | Default | 111/0 kb

MetaTrader installation 1

Bars in test	7041	Ticks modelled	25522812	Modelling quality	n/a
Mismatched charts errors	0				
Initial deposit	10000.00			Spread	Current (19)
Total net profit	123.80	Gross profit	4789.70	Gross loss	-4665.90
Profit factor	1.03	Expected payoff	0.45		
Absolute drawdown	497.60	Maximal drawdown	694.70 (6.81%)	Relative drawdown	6.81% (694.70)
Total trades	276	Short positions (won %)	138 (28.26%)	Long positions (won %)	138 (34.78%)
		Profit trades (% of total)	87 (31.52%)	Loss trades (% of total)	189 (68.48%)
	Largest	profit trade	244.80	loss trade	-180.60
	Average	profit trade	55.05	loss trade	-24.69
	Maximum	consecutive wins (profit in money)	5 (229.70)	consecutive losses (loss in money)	10 (-194.50)
	Maximal	consecutive profit (count of wins)	244.80 (1)	consecutive loss (count of losses)	-348.70 (9)
	Average	consecutive wins	2	consecutive losses	3

For Help, press F1 | Default | 66/0 kb

MetaTrader installation 2

Above is a screenshot of two different MT4 installations testing the same strategy on the same data. First has modeling quality 99%, second has n/a, but as you can see the ticks modelled and strategy results are exactly the same.

To sum it up – **you don't need to worry about n/a modeling quality if there are no chart errors and the whole bar is green.**

7 Troubleshooting & Recommendations

7.1 When nothing gets build for a long time

Have you started Builder and no strategy appears in databank for a long time?

It depends on your setting, if you set testing precision too high or complex cross checks and filtering, it might take a lot of time to generate strategies that pass all the filters.

But normally, you should see new strategies added into databank every few seconds or minutes. If no strategies are added into databank for a very long time there might be some problem with your configuration.

There might be few different reasons:

7.1.1 Using Genetic evolution with too big population

Initial population in Genetic evolution serves as a starting point of evolution – it is not stored into databank. If you set it wrong, you might end up with SQ spending hours or even days generating strategies only to create this initial population, before it even gets to evolution.

The screenshot shows the 'Builder' interface with the 'Settings' tab selected. Under 'Advanced settings', the 'Genetic options' tab is active. The 'Genetic options' section shows 'Max # of Generations' at 30 and 'Population Size (per island)' at 1000. The 'Islands options' section shows 'Islands (separate evolution)' at 8, 'Migrate every Xth generation, X =' at 50, and 'Population migration rate' at 15%. The 'Initial population generation' section has 'Use strategies from Initial population databank as evolution start' checked, 'Initial population size' at 8000, and 'Generated decimation coefficient' at 2. The 'Filter generated initial population' table is as follows:

Use	Left value	<=>	Right value	
<input checked="" type="checkbox"/>	# of trades	>	1000	×
<input checked="" type="checkbox"/>	Net profit (IS)	>	-5000	×
<input type="checkbox"/>	Avg. Bars in Trade	>=	3	×

On the screenshot above you can see that there are 8 islands x 1000 population on each, which means 8000 strategies for initial population.

Moreover, it has Decimation set to 2, which means that it will generate double this amount of strategies and chooses the best 8000 from them.

So SQ must generate 16.000 strategies that have to pass the Initial population filter.

This task alone could take hours or days.

Recommendation:

Think about your genetic settings, start with smaller population, and Decimation=1. Also, monitor your rejection statistics – isn't your Initial population filter too strict? You can try to use Random generation first, to see how quickly it generates strategies with these conditions.

7.1.2 Using Genetic evolution with too strict filters

Similar problem as the previous one – you let generation run for some time, but no strategy passes the filters. This probably means that filters are either configured incorrectly or they are too strict.

Strategies generated:	515		
Time per strategy [Detailed] :	359 ms.	Time per accepted strategy:	0 ms.
Rejected [Detailed] :	5 / 100.00 %	Accepted:	0 / 0.00 %
Strategies per hour:	10 027	Accepted strategies per hour:	0.00

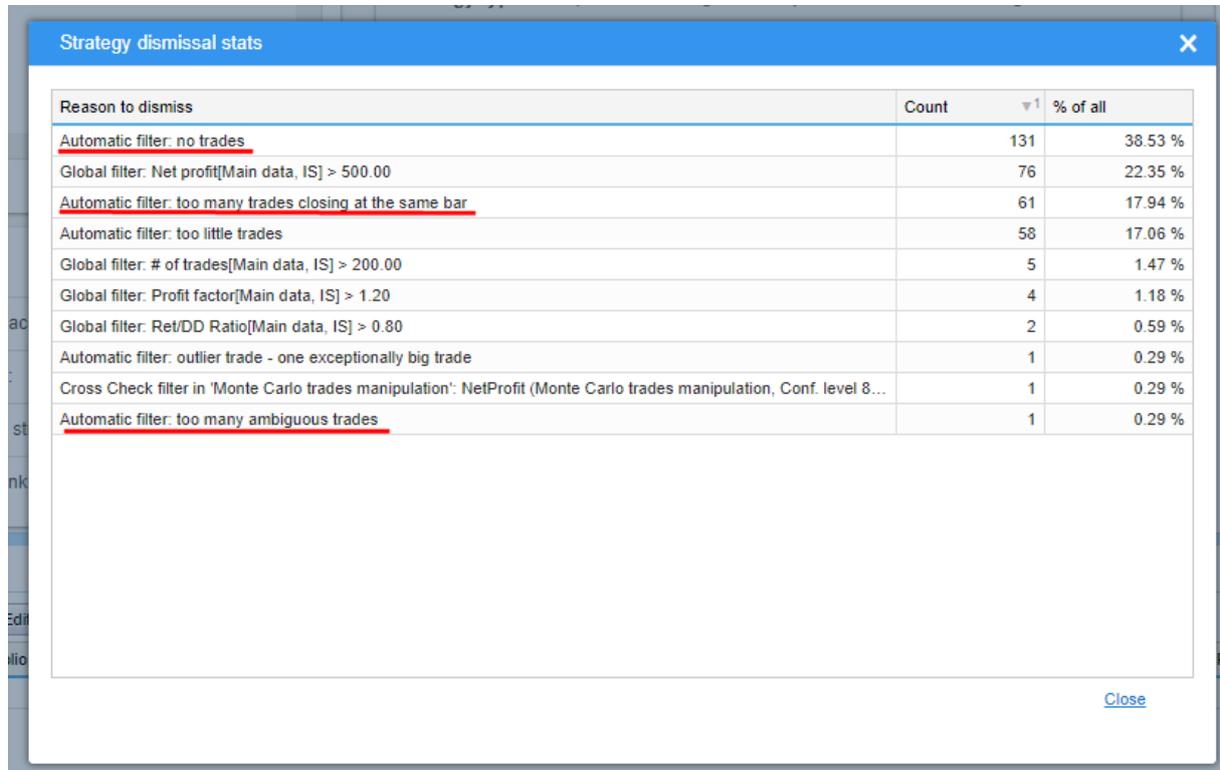
You can check the detailed Rejection statistics to see why the strategies are getting rejected, but good guess is that the Initial population filter is too strict.

Recommendation:

Try to use Random generation first, with the same filter settings, to see how quickly it generates strategies with these conditions. If it takes too long then there might be some problem with settings and you might need to alter it.

7.1.3 Getting too many rejections based on Automatic filtering

Automatic filters in SQ serve to filter out strategies with obvious flaws. You should check the Rejection statistics to see if there aren't too many strategies rejected because of this.



Reason to dismiss	Count	% of all
<u>Automatic filter: no trades</u>	131	38.53 %
Global filter: Net profit[Main data, IS] > 500.00	76	22.35 %
<u>Automatic filter: too many trades closing at the same bar</u>	61	17.94 %
Automatic filter: too little trades	58	17.06 %
Global filter: # of trades[Main data, IS] > 200.00	5	1.47 %
Global filter: Profit factor[Main data, IS] > 1.20	4	1.18 %
Global filter: Ret/DD Ratio[Main data, IS] > 0.80	2	0.59 %
Automatic filter: outlier trade - one exceptionally big trade	1	0.29 %
Cross Check filter in 'Monte Carlo trades manipulation': NetProfit (Monte Carlo trades manipulation, Conf. level 8...	1	0.29 %
<u>Automatic filter: too many ambiguous trades</u>	1	0.29 %

There are few very common rejection reason and their causes:

Automatic filter: No trades

This simply means that generated strategy doesn't trade at all. Why is that? Very probably the strategy conditions are generated in such a way that they are never true.

An example of no trades strategy:

```
LongEntrySignal = (((BearsPower(Main chart,36) > 10.0)
  and (AwesomeOscillator(Main chart,) crosses 0.0 upwards))
  and (RSI(Main chart,20)[3] crosses below 75))
  and Ichimoku(Main chart,9, 26, 52) price crosses KijunSen bearish);
```

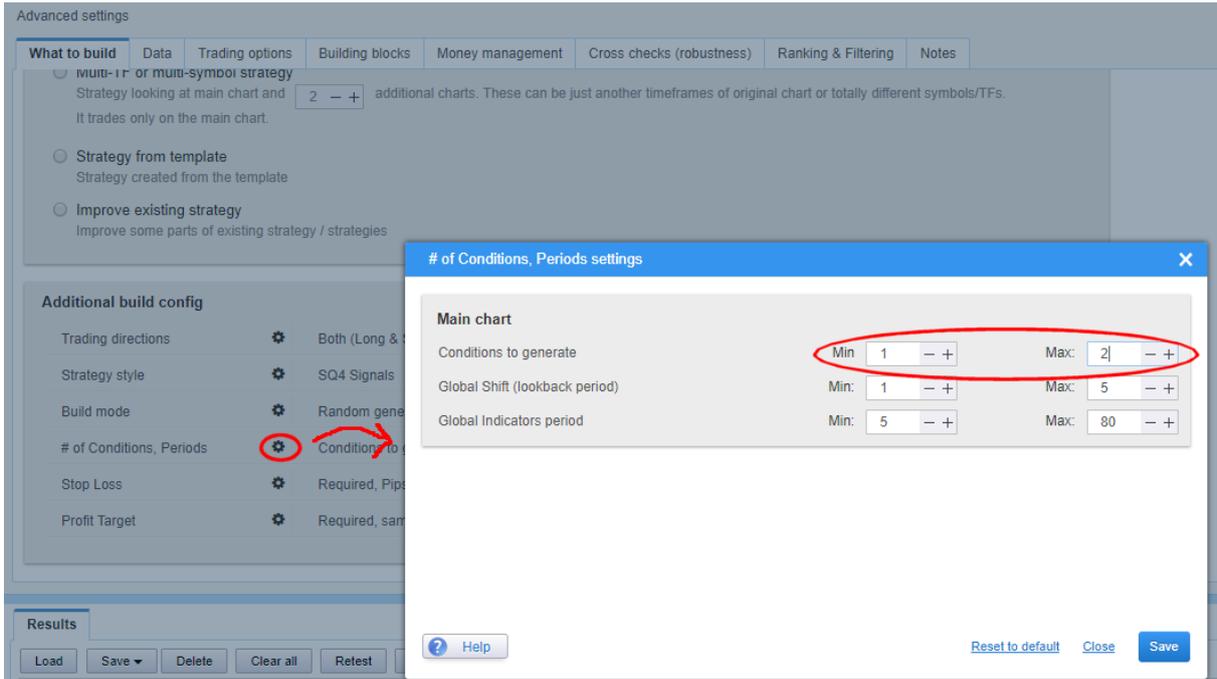
```
ShortEntrySignal = (((BearsPower(Main chart,36) < 10)
  and (AwesomeOscillator(Main chart,) crosses 0 downwards))
  and (RSI(Main chart,20)[3] crosses above 75))
  and Ichimoku(Main chart,9, 26, 52) price crosses KijunSen bullish);
```

This strategy was generated with 4 conditions for Long and Short signal, and it seems that they are never true at the same time.

Recommendation:

Configure SQ to generate less conditions. The more conditions you let it generate the more prone the strategy will be to curve fitting, and the more time you'll see no trades problem. It is recommended to use 1 to maximum 2 conditions.

How to configure it? Go to **Settings** -> **What to build** and there edit # of Conditions. In the opened dialog set the maximum to 1 or 2.



Automatic filter: Too many trades closing at the same bar / ambiguous trades

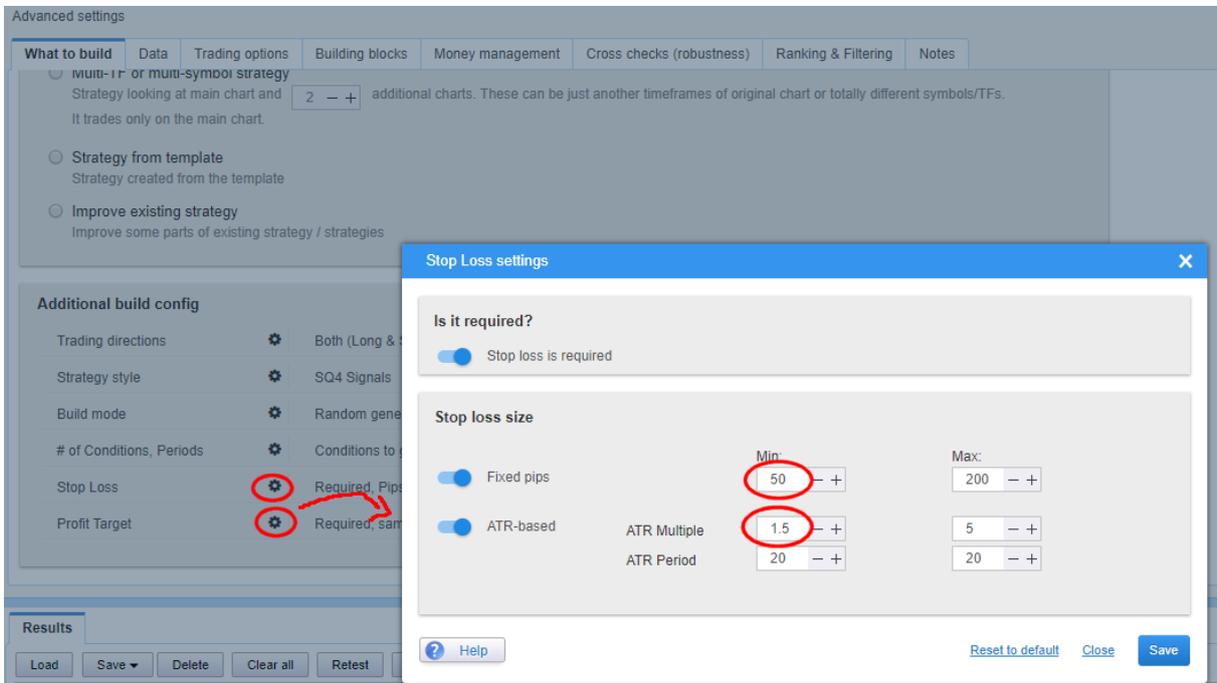
Another common problem is when trades open and close at the same bar. This is a problem, because such strategy cannot be reliably backtested.

It again suggests problems with settings. Most probable cause is that your generated Stop Loss and Profit Target values are too small.

Recommendation:

Configure SQ to generate bigger SL and PT. The correct size depends on your market and timeframe, so if you use fixed values you should configure its min and max according to this. If you use ATR-based values, use at least 1.5 as the minimum.

How to configure it? Go to **Settings -> What to build** and there edit Stop Loss and Profit Target. In the opened dialog use at least 1.5 for ATR Multiple minimum and appropriate value for fixed pips minimum.



8 Appendix

8.1 Ranking criteria

TO BE ACTUALIZED FOR SQ X

Net Profit

Net Profit is a total profit/loss the strategy produced.

Drawdown, Max DD %

Drawdown is the measure of the decline from a historical peak in running cumulative profit of the strategy. Max DD % is the maximum percentage drawdown of the strategy.

Stagnation, % Stagnation

stagnation is a maximum number of days during which strategy stagnates - which means it doesn't make a new high on the equity. Obviously you'd want strategy with as little stagnation as possible.

% of Wins

percentage of winning trades

Annual % Return

average annual percentage return of the strategy. If it is 30 %, it means the strategy makes 30% every year in average.

Annual % Return / Max DD %

important ratio between annual percentage and maximum percentage drawdown. You'd want this number to be as high as possible.

Avg Daily, Monthly, Yearly profit

average profit for given period in \$

Return/DD Ratio

a very good measure of strategy quality, the higher the number, the bigger are profits in relation to maximum drawdown.

Expectancy

performance metrics developed by Van Tharp, it is computed as (percentage wins * average win) - (percentage losses * average loss)

R-Expectancy

performance metrics developed by Van Tharp, it gives you the average profit value related to average risk (R) that you can expect from a system over many trades.

You can find more at: <http://www.vantharp.com/tharp-concepts/expectancy.asp>

R-Expectancy Score

standard R-Expectancy doesn't consider the length of testing period and number of trades produced. There is a difference when you make for example \$2000 per year by making 10 trades, or by making 100 trades.

R-Expectancy Score adds a score for trades frequency. It is computed as: $R\text{-Expectancy} * \text{averageTradesPerYear}$

SQN (System Quality Number)

performance metrics developed by Van Tharp, it is the measure of the quality of a trading system.

You can find more at: <http://www.vantharp.com/tharp-concepts/sqn.asp>

Standard interpretation of SQN is:

- Score: 1.6 – 1.9 Below average, but trade-able
- Score: 2.0 – 2.4 Average
- Score: 2.5 – 2.9 Good
- Score: 3.0 – 5.0 Excellent
- Score: 5.1 – 6.9 Superb
- Score: 7.0 - Keep this up, and you may have the Holy Grail.

SQN Score (System Quality Number Score)

just like in case of R-Expectancy, SQN doesn't consider the length of testing period and number of trades produced. In fact it is more favorable for systems that produce more trades, without considering the length of the testing period.

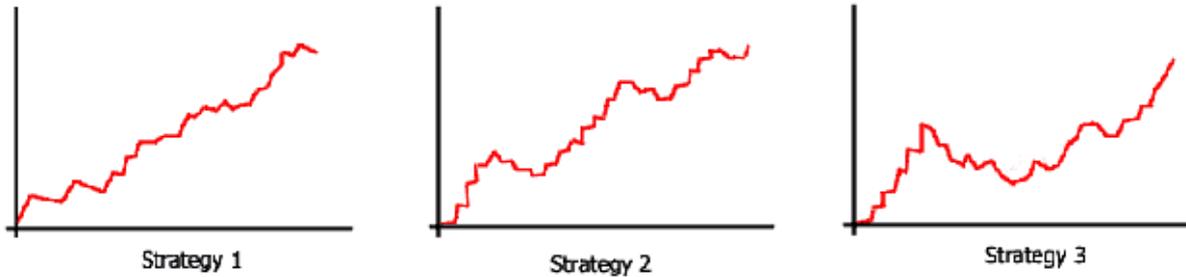
It is computed as: $SQN * (\text{averageTradesPerYear} / 100)$

Number of trades

simply a number of trades of this strategy in the backtest. You can use this criterion to approximate some preset value you would like to achieve (for example total 100 trades). The value of this criterion will be bigger for strategies closer to the desired number of trades.

Stability

a special value in range from 0 (worst) to 1 (best) that measures how stable is growth of the equity chart.



You can see three sample strategies on the picture above. All of them finished with the same profit, but Strategy 1 was growing almost linearly (very stable), Strategy 2 was growing with occasional big drawdowns (less stable) and Strategy 3 was moving up and down (very little stability of growth).

Stability is a value that is quite good in representing a "quality" of the trading strategy and it can be used as the only one or the main criterion to compute the total strategy *Fitness*.

Symmetry

criterion to maximize strategy symmetry. Symmetry value is in %, and it is measuring how much is the Profit/Loss for Long direction similar to Short direction.

For example, if strategy makes \$600 on Long trades, and \$400 on Short trades, symmetry in this case is 66%. (\$400 is 66% from \$600).

If the strategy makes the same profit on both directions, the symmetry will be 100%. If one of the directions produces loss or 0 profit, the symmetry will be 0%.

Win/Loss Ratio

criterion to maximize the ratio of winning trades vs. losing ones

Average Win, Average Loss

criterion to maximize average win or minimize average loss per trade

Average Bars in Trade

criterion to minimize the average number of bars the trade is open

Average Bars Win, Average Bars Loss

criterion to minimize the average number of bars for the winning/losing trade

Degrees of Freedom

in a trading system relates to the number of criteria that are used to filter price action and/or volume and determine entry points.

The more criteria and variables used to determine entry timing points, the fewer degrees of freedom the system will have and vice versa.

Degrees of freedom are computed from strategy complexity and number of trades. The simpler the strategy is, the more degrees of freedom it will have. For this property, the bigger value is better.

Complexity

measures the complexity of the strategy. It is simply a count of all the indicators, prices, operators and other building blocks that are used in the strategy. The higher this number is, the more “complicated” the strategy is.

For this property, the smaller value is better.

Both properties can be used also in the ranking to influence the Fitness function.

8.2 Entry and exit types

TO BE ACTUALIZED FOR SQ X

Profit Target (PT)

is the opposite of the Stop Loss, it can be used to take the profit once the price moves in your favor. As in the example above, you can set up a profit target to close the position once it makes for example 100 pips.

Exit After X Bars

this is very simple exit condition. If used, the strategy will close the trade after given number of bars (time periods on a given timeframe).

This is many times a simple, but effective way to close the trade when the market moves sideways or when we want to lock in the profit.

Move Stop Loss to Break-Even

another simple but quite powerful condition. It tells the strategy to move protective Stop Loss to the entry price (break-even) when the given profit is reached. This will not close the trade, so the strategy will be still in position, but there is zero risk after this move. Because SL is at the entry price we'll not lose anything even if the trade reverses and goes against us.

This condition is evaluated only on open bar (not on every tick).

Profit Trailing

a simple trailing stop that trails the Stop Loss specified distance from highest achieved profit. This condition is evaluated on open bar (not on every tick).

Stop Trailing

more advanced trailing stop that can use a value of indicator or price (Open, High, Low, Close) to trail the Stop Loss. For example the rule can trail SL at $\text{Lowest}(20) + 20$ pips.

This condition is evaluated on open bar (not on every tick).

Exit Rule (Price + Operators + Indicators, ...)

it is a special type of exit where strategy has not only rules for entering the trade (Entry rules) but also rules for exiting it (Exit rules). Exit rules look as same as entry rules - it is a combination of all the selected building blocks(indicators, operators, price values).

Example of entry and exit rule pair:

Entry Rule - if $\text{CCI} > 0$ Enter at Market

Exit Rule - if RSI crosses above 100 and we are in long position then Close Trade

Note on Stop Loss / Profit Target/Move Stop Loss to BE

Both Stop Loss and Profit Target can have either fixed value in pips (like 30, 50 or 100 pips), or they can be based on ATR (Average True Range), for example $2 \cdot \text{ATR}$.

This means that the actual value of the SL/PT will be 2 times the value of average true range at the time the trade was opened.

ATR based SL/PT have the advantage that they "adapt" to changed market conditions and changes of volatility. Some strategies (but not all of them) can be more profitable when they use ATR based SL/PT instead of fixed one.

To enable/disable ATR for SL/PT, you can enable/disable the Average True Range indicator in the Entry Building Blocks.

9 Final Words

Congratulations on completing the guide to StrategyQuant. I hope you'll like the program and you'll use it as an important part of your trading.

StrategyQuant is very new program, and it is underactive development. This means you can look forward for new exciting features that will move the program functionality even further.

I'm open to a new ideas or suggestions, so if you are missing something in StrategyQuant or if you think something can work in a different way, don't hesitate to let me know.

I wish you many successful trades,

Mark Fric

P.S. I would love to hear your success stories, so please let me know at <mailto:support@strategyquant.com>