

DATFRA - Mt4 EA Builder & Walk Forward Analyser



Darwin-FX | Joined Oct 2013

First Post: Jul 10, 2014 8:02am

[Quote](#)

[Cleanup](#)

[Post 1](#)

Hello fellow traders 😊

Today I can finally release my algorithmic trading framework. I have been working on this for many months, and now its time to share it.

Its an algotrading framework, which builds on top of Metatrader4 and MQL Expert Advisors, and it is meant to analyse, optimise and manage our EAs.

Features:

- **EA Builder:** A simple to use EA builder allows you to let your computer search for new trading systems - without human interaction
- **Walk Forward Analyzer:** Analyse your trading systems with Walk Forward Analysis - fast, effective, multithreaded.
- **Metatrader4 Integration:** DATFRA works on top of Metatrader4, you do not have to change your preferred trading enviroment to use it!
- **MQL Library:** Many MQL Functions and EA templates come with DATFRA, everything you need to immediately start EA programming!
- **System Management:** The internal Database allows you to manage your EAs, parametersets and various kinds of analysis reports - neat and central
- **Data Manager:** DATFRA can import history data from Metatrader4, Metatrader5 or CSV files.

Planned Features: (lot of them are already coded):

- **Parameterspace Analysis:** An analysis method 10.000 times as powerful as Walk Forward Analysis, read more about its concepts: <http://darwins-fx-tools.com/?articles&art4>
- **Professional & Flexible Builder:** This will allow you to use any MQL Function, any Metatrader indicator and any Expert Advisor for system building - full freedom for professional traders
- **Portfolio Creation:** Analyse correlation of trading systems - and build the ideal portfolio
- **Live Optimisation:** Let DATFRA re-optimize your EAs for the current market conditions - based on your findings during walk forward- or parameterspace analysis.
- **Cloud Builder:** The simple EA Builder - which is already implemented - will upload it's found EAs, for the whole community to download. Let's work together finding the best trading systems
- **Monte Carlo Simulations:** Determine the real statistical characteristics of a trade-distribution

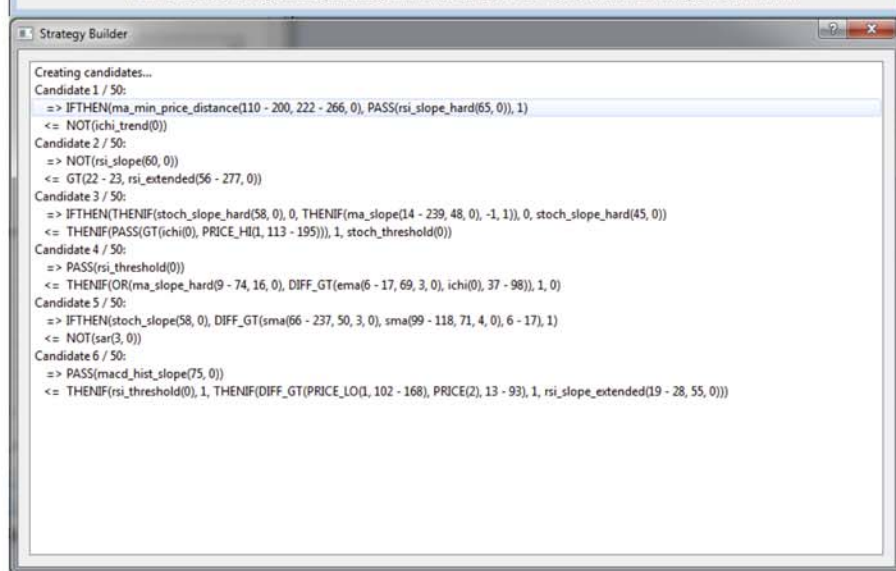
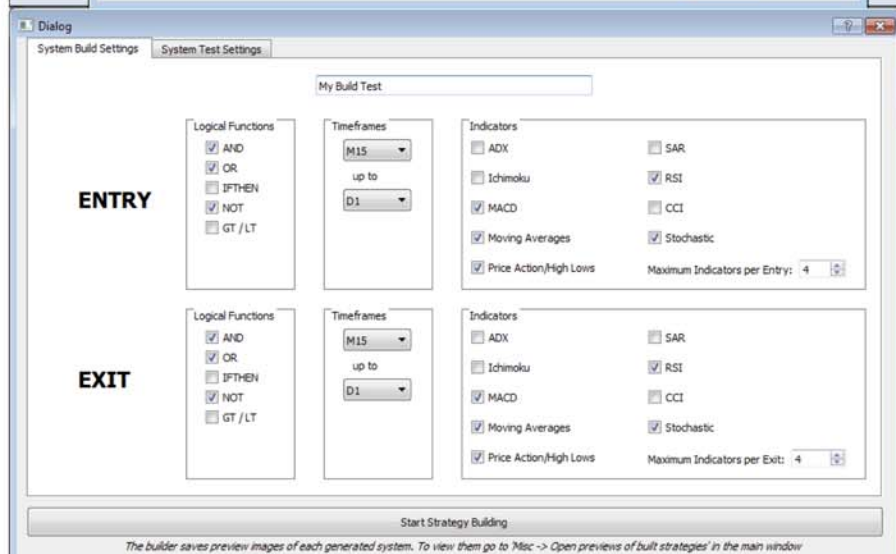
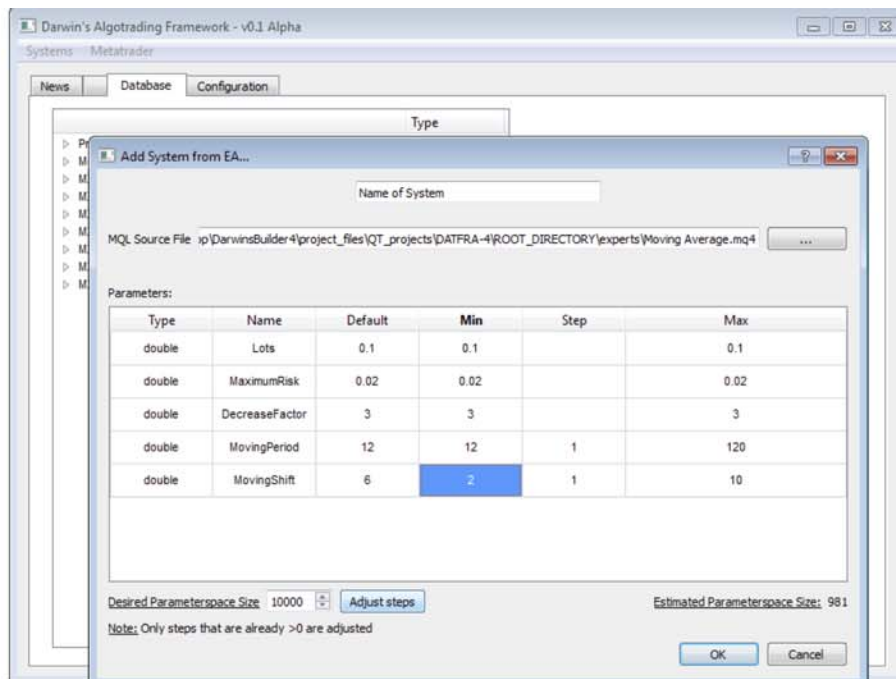
This is an **alpha version**, so please be so kind and report all bugs to me via skype, pm or e-mail: <http://darwins-fx-tools.com/?contact>

As long as this is a tiny project, everybody is also invited to contact me for general help or questions about the framework.

The software is **free** - and I want to keep it that way. However, I am not just sharing this because I am such a nice person, but also because I want to make some valuable contacts and reputation through this project. So, if you like it, pay back through sharing this with as many traders as possible 😊

As this is under heavy development, I will update this thread frequently, whenever I change something or add new features - so subscribe if you do not want to miss it!

Screenshots:



Walk Forward Analysis Settings

General Options **Advanced Options**

Markets
 EURCAD
 M30
 Add this market to list >>

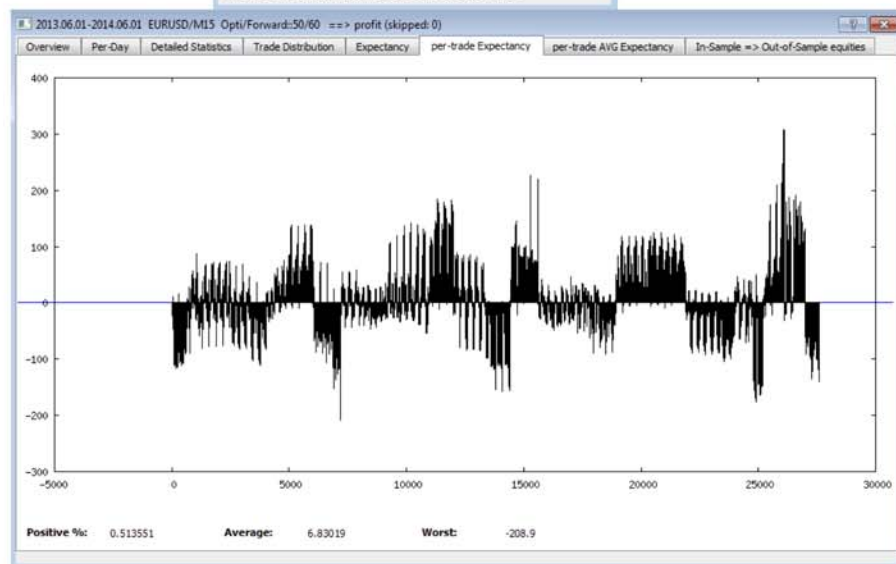
Data to use
 History: 12.08.2011 - 01.06.2014
 Spread: 3
 Due to optimisation, trading will start: 2012.08.11 00:00
 Subtract optimisation time from start

Optimisation Config
 Guide optimisation towards: Genetic - Profit
 Preferred Characteristic: profit

Timespans
 Optimise on: 365 days of data,
 trade for max: 60 days before reoptimisation.
 Skip: 0 % of all wf-windows.

Start on all markets (one by one)

Attention! WFA is highly random and unreliable, due to its nature!



Download & Instructions:

Download the tool here: <http://darwins-fx-tools.com>

ATTENTION: There is a manual linked on my website, READ IT ALL! You will not even be able to install this software if you do not carefully follow the instructions (thats not a 'just click on Next' kind of installation routine!)

The manual is constructed out of slideshows with a bit of text, so it wont take long. But you MUST read every text below every image, there are sometimes very important instructions written there.

If you do not do so - please, dont bother using my software.

-Darwin

Last Post: Jul 16, 2

Edit

Quote

Cleanup

Post 2

Why backtests are useless, EAs are flawed and their parameters are bad

THE GENERAL APPROACH TO SYSTEMATIC TRADING

Take a trading logic => Generate an EA that trades exactly like it => Optimise Parameters => Do a backtest => Trade it live => \$\$\$

BUT: Everybody that has ever had an EA that was a money-printing machine in the backtests (and this is very, very easy to be done), knows that a good backtest does not imply profitable live trading. And that is the problem. We need a test-method on which evaluation we can rely!

STATEMENT 1: Your parameter choices are not good

Well, every EA consists of 2 things:

1. the logic/script itself
2. the parameters (like periods for moving averages, stop loss values etc. Just everything that can be adjusted!)

The first thing is static and "given". And a lot traders only focus on this part.

But, an EA can behave in very different ways, depending on the parameters. And there can be billions of possible parameter-combinations (=parameterset).

There are 3 approaches to determine the parameterset for live trading:

1. the intuitive approach (non adapting):

The trader just chooses the parameters based on his expert knowledge. But, there are just too many possible parametersets. You can't just "guess" them without evaluation and testing. It does not matter how good you are, you are never good enough to "think through" this huge amount of possibilities in a reliable fashion!

2. the "optimise on all data" approach (non adapting):

The trader chooses the parameters based on all past data. So, the configuration/parameters that worked best for the last 12 years, for example, are chosen to be traded live. But, to stay up-to-date it is not a very good method to optimise the parameters on so much and therefore OLD data. Because the market today is not the same as it was 12 years ago.

3. the "optimise on the last few years" approach:

The trader chooses the parameters based on the last few years of data. But you can not just take this approach without testing how this "optimisation method" would have worked in the past. And this is exactly what a walk forward analysis will do: "Optimise on the last few years", but it tests this approach on all data in the past!

STATEMENT 2: A non-adapting EA can never make longterm profits

I mean, there is an infinite amount of possible trading-systems AND an infinite amount of ways the market can change AND an "infinite" amount of possible parameters for your trading-system.

Considering this, would you really want to bet money on the fact that you have a trading system that will always work, in the same way, without adaption, on all future market conditions? I would not.

"But why shouldn't I be able to put a traders knowledge into a script and trade it?"

Its simple: A trader always learns, he takes input from many sources, he has knowledge about the markets and therefore he adapts his trading strategy so he can always be as close to the markets as possible.

A simple EA script can't do this, a Walk Forward Analysis can (to some extent).

Reason #1 why there are many "profitable" EAs out there

The EA really works, it has a sound strategy and was developed properly. But the markets always change, and they can change in infinite ways.

So, at some point, the traded market-inefficiency WILL(!) change, and the EA can not adapt to this change and therefore will lose its profitability.

Reason #2 why there are many "profitable" EAs out there

is that some of them, sound or unsound systems, are just lucky.

If you send 10.000 people to a casino and let them play for a while, just due to chance, some of them will make profit over some time.

The same is for EAs, if there are enough of them, some will really make profit (even if they are in fact useless).

But as they keep trading, they will lose, as the probabilities are against them (same as in a casino).

Reason #3 why there are many "profitable" EAs out there

is the huge risk some EAs take (grid trading, martingale systems etc).

These EAs take very huge risks to make small profits. But somewhen, the risk strikes, and at that point, the EA will to lose (all) money.

Reason #4 why there are many "profitable" EAs out there

is the small timespan on which they are profitable. Its no magic to make profit for months, but its hard to make it for many, many years.

STATEMENT 3: A backtest does not tell you anything about the future performance

Well, don't get me wrong, I strongly hold the opinion that simulations on the past are the only way to really test a trading system.

But what does a backtest tell you? Just that your system performed well on the past.

But is trading really about having a system that represents the past?

No, trading is about designing a system on the past and then trade it in the future.

THIS IS A FUNDAMENTAL DIFFERENT QUESTION THAN WHAT A BACKTEST CAN TELL YOU!

You want to answer the question "how good is my live trading performance" with the answer to the question "how good did my system perform in the past". That logic is flawed, of course.

"But if I use out-of-sample data to verify my backtest..."

Out of sample testing is a good idea. But you only have one optimisation-dataset and one test-dataset, which is not very reliable.

Walk forward analysis, somehow, is out-of-sample testing on steroids. It uses the same method, but generates 10-1000 opti/test dataset-pairs.

"So, are all backtests useless?"

No, it can help you to get sound and good trading systems, if done right.

But my point is that a good performance in a backtest does not make sure the system is also good in live trading.

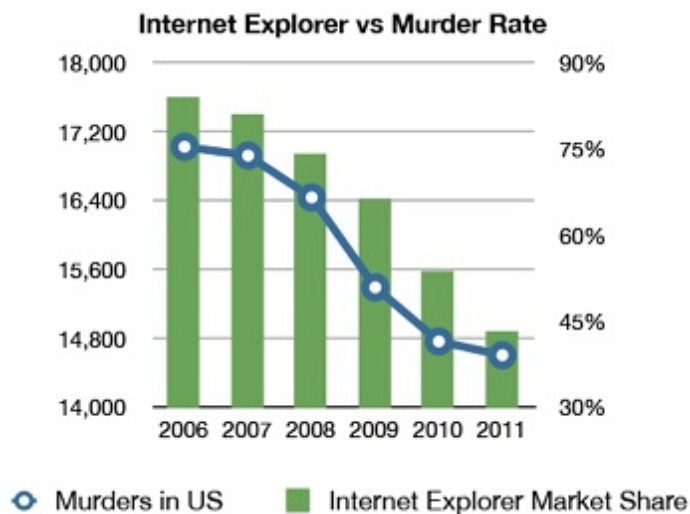
Trading is a game of probabilities, and the probability that an EA is profitable, only based on a good backtest, is very, very low.

"But why should a system that performed good over the past simply stop working"

The main reason is, and that might be the most important one, OVERFITTING.

That means, the EA tackles an "random" inefficiency of the market that was just there in the past, but that was no real thing.

I found a very good picture that explains 'overfitting':



You see? There are infinite of such useless "relationships" in the price data, and most EAs tackle one of them which gives them very good backtest results, but that was just due to luck, and these "relationships" will not hold in the past.

And as there are a lot more of these overfitted solutions than sound ones, the chances are very high that you optimised towards such an overfitted system.

The other reason is that the markets always change. So even if you have a non-overfitted, good performing backtest, you will run into several problems that you can read in Statement 2

THE SOLUTION: Walk Forward Analysis

Well, some of you will already know how a walk forward analysis works. For the others here is a short overview, as I will explain the process in more details in a future thread.

A WFA takes, for example, the data from 2000 to optimise your system, then tests it on 2001 (which is, from the point of optimisation, the "future" or "live trading").

Then it walks forward and optimises the system on 2001, test it on 2002. Then optimise on 2002, test on 2003 etc.

Do this until you walked through your whole data and only consider the "live trading" in your evaluation.

You see? With this simple tactic you can tackle the 3 problems I have described above, the lack of adaptability, the need for a evaluated parameter selection process and the uselessness of a "past-performance"-backtest.

And then, when you want to trade the system live, you do the exact same thing that you have tested:

- 1.) Optimise on the last available data, with the same optimisation settings you tested in WFA
- 2.) Trade it in the future, with the same procedure you tested in WFA

And that is the main difference to normal backtests: You trade your EA in the exact way you traded it

[Learn more about how Walk Forward Analysis works](#)

Walk Forward Analysis - the only logical successor to backtesting

Initial Situation

First, let's sum up what the essential parts of every trading system are!

1. The system's logic

The most obvious part! And for a lot beginners it's the only part they know, which is dangerous.

This might be a manual trading system or an expert advisor or any other form of fixed trading-logic / trading-system / trading-strategy (btw: all 3 terms name the same thing in this article)

So far, so good. But you all know that every strategy has some kind of variables/parameters (like the periods of moving averages or stop-loss levels etc), that are NOT FIXED(!) but can vary, which brings us to the second part. (If you just set them to a fixed value because "this should work".. well, it wont, at least not in the long term)

2. The system's parameter-ranges

The ranges of the parameters are an ESSENTIAL part of every trading-system, as they determine the exact behaviour of it (tough the trading-logic always stays the same).

So, for example, a moving-average-period might range from 5-15 to capture short-term price movements.

It is not 6 and not 11 and not 14, it is 5-15, as the markets change, we can not choose a concrete value, ALWAYS a range!

3. The market, the amount of data, the desired characteristics

Every strategy trades on a market, so we want to determine on which. (eg EURUSD / H4)

But thats not enough, we also have to determine how many past-price-data we want to use to evaluate our possible parameter-choices.

Because, as I said, a system always has parameter-ranges, but for live trading we have to choose concrete values!

And we do this by evaluating all parameter-possibilities on the last X years of price-data, and after evaluation, we end up with a huge list of possible and "independent" trading-systems (each of them with different parameters, but

the same main-logic).

And each has its own characteristics like "profit" or "profit factor" or "relative drawdown".

So, we also have to determine how to pick the "best" parameters.

But it's not as simple as saying "I want much profit", because the characteristics often don't hold in the future!

Instead we want to choose in a way that gives us a high probability of picking parameters that will succeed during live trading.

Simple, isn't it?

An illustrating example

The system's logic:

Let's suppose a very basic trading-system: "If the price moved more than X pips in the last Y days, a course correction will happen" (this is not a valid strategy, it's just thin air for the sake of simplicity).

The parameters would be X and Y in this case.

The system's parameter-ranges:

To make it all simple, I chose X to be 100-200 pips in this example, and Y to be 2-3 days. (also, just thin air!)

Amount of data & preferred characteristics:

Here we choose the last 10 years of data to evaluate the possible parameters on, and "profit" as preferred characteristic.

(tough, as I said, in reality profit is not a very good indicator for parameters that will have a good future performance)

The process:

Ok, now before we can trade that system, we make an optimisation on the last 10 years.

That means we backtest every possible parameter-combination for our system and choose the best in terms of "profit".

For sake of simplicity, here is a cropped example:

"If the price moved more than 100 pips in the last 2 days, a course correction will happen". => 1000\$ in the last 10 years.

"If the price moved more than 150 pips in the last 2 days, a course correction will happen". => 1200\$ in the last 10 years.

"If the price moved more than 200 pips in the last 2 days, a course correction will happen". => 1500\$ in the last 10 years.

"If the price moved more than 100 pips in the last 3 days, a course correction will happen". => 900\$ in the last 10 years.

"If the price moved more than 150 pips in the last 3 days, a course correction will happen". => 950\$ in the last 10 years.

"If the price moved more than 200 pips in the last 3 days, a course correction will happen". => 950\$ in the last 10 years.

Soo, According to our preferred-characteristic (profit), we would choose $X = 200\text{ pips}$; $Y = 2\text{ days}$, and then just trade the strategy.

Well, that's the "normal" process of EA-trading, and I claim it does not work this way.

EA-Analysis: How to ask the right questions

Ok, now that I have described the current process and how it is all done, here comes the "new" part.

The goal itself always stays the same, we want to pick the best parameters (based on some kind of evaluation on the past), and then we want to trade live!

Remember: The only thing a backtest can tell is "How good did my system+parameters perform in the past".

But that is NOT what we want to know! Be sure that you really understand this.

Initial Question; What we actually want to prove with analysis.

"Does the way we choose parameters for live trading ('pick the one with best profit over the last 10 years' in the above example) give us a high probability to pick parameters that are profitable during live trading?"

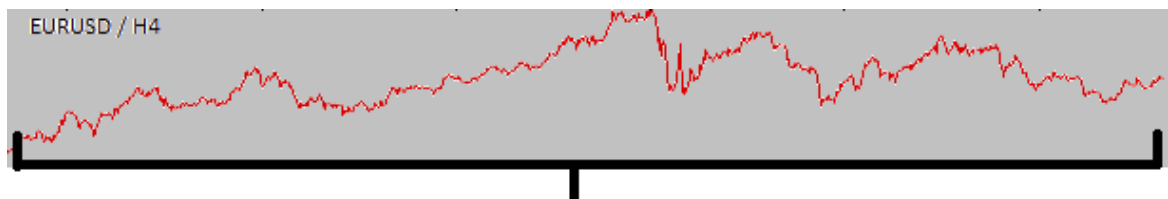
So we are actually interested in the relationship of past-performance&future-performance, not backtest-results!!

If the answer is **No**, one or more of the 3 things described in "Initial Situation" are wrong. Might be the logic itself, the parameterranges etc..

If the answer is **Yes**, the performance in the past and the performance in the future are somehow correlated for our EA, and we can trade the system!

The logical evolution; From Backtests to Walk-Forward-Analysis

Step One: Backtesting - in it's worst form



We seek the best parameters on the whole available data, then we take the performance on this data as indicator for future profits

Pro:

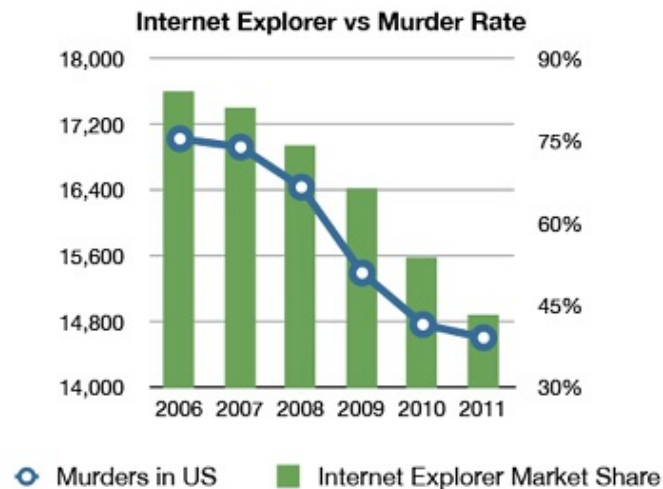
- We will get parameters that performed well on a wide range of data

Contra:

- Every single EA trader that used this method and then tried to trade an EA live, based on good backtests, can tell you: It just does not work this way.
- Overfitting / Curvefitting! We first optimise the parameters, and then test them, all on the same data. That means we have no clue if we have valid parameters or overfitted ones.

Overfitting means, we optimised towards a random behaviour within our data, that just exists in this particular dataset, and will not hold in the future.

That means, we captured a relationship that existed but was not a sound one. Like this:



Don't fool yourself in thinking "ah, this wont happen"... Almost all "relationships" within the markets are like this, as most price-movements are random!

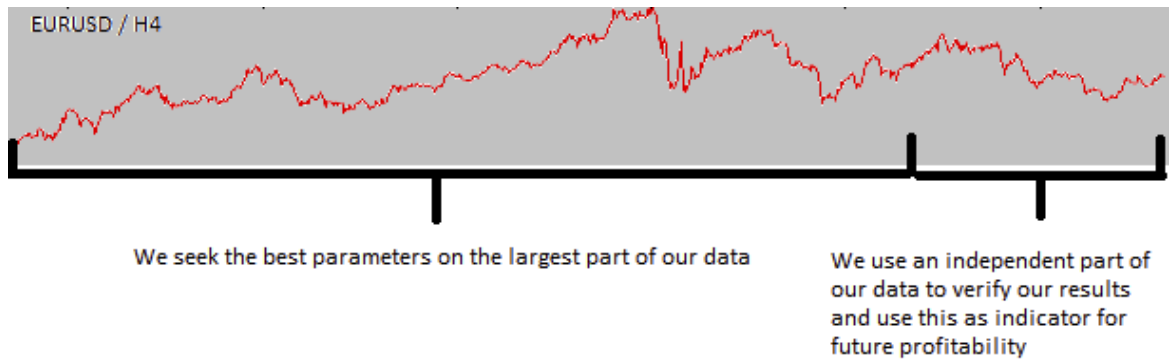
If you do not understand overfitting, google for more information, as it is our archenemy in mechanical trading.

- No significance for future performance! Remember that the initial question is not how good our parameters performed on the past, but how high the probability of succes AFTER the optimisation-timespan (so, "in the future", during live trading) will be. As we did not do any tests with our parameters that take into account the relative future, we did not even try to answer the initial question.

We just answered the question "How good did our parameters perform in the past", not taking into account anything about the "future" => very bad.

- Even if you could somehow magically invalidate my above points, because the parameters worked well on a huge amount of data, they are not really the best for the current market - just average good on all market-conditions.

Step Two: Backtesting using unseen/out-of-sample data



Notice: The first dataset, we use to optimise our parameters on, is called "in-sample" (is). The second, unseen, dataset is called "out-of-sample" (oos).

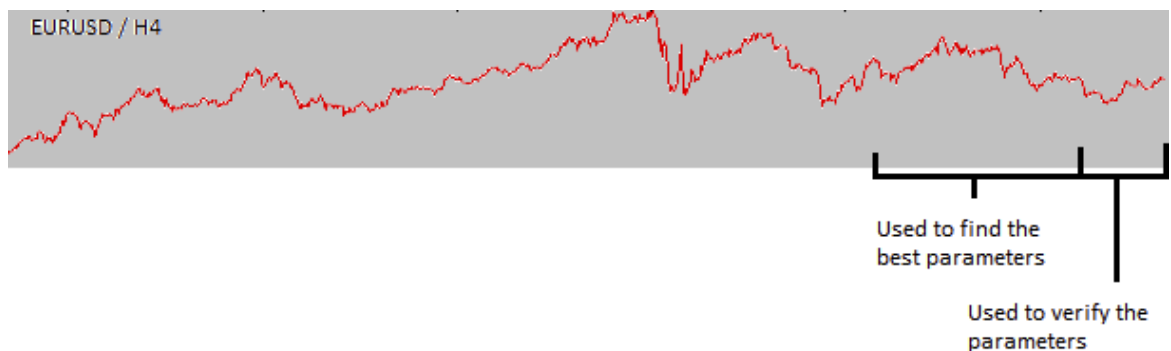
Pro:

- We now have a lower chance to get overfitted parameters, as we use an independent dataset to validate our parameter-choices.

Contra:

- Due to the infinite amount of senseless/unsound relationships within the markets, we still have a (too high) risk for overfitting, as chances are too high that we just got parameters that are valid (curvefitted) on both datasets, but not valid in the future.
- If the system did not work in out-of-sample and you then begin to tune your parameters until you get good oos-results, your oos-results are not longer "unseen" and becoming "in-sample", which makes the whole approach using 2 datasets useless!
- We still use a very large part of our data (in-sample) to find the best parameters, which also means we use a lot "old" data. That is not a good decision as the behaviour of the markets in the past is not equal to the behaviour of today.
- Not just our in-sample dataset is too huge, also our out-of-sample dataset is too huge and therefore un-realistic. In the example above it would be a few years, but would you really like to trade a system for years before choosing new, re-adjusted, parameters? I would not!

Step Three: Backtesting using a more realistic data-amount



Pro:

- We now only use the recent market-behaviour to optimise our parameters, so we capture the market "at the moment", and not "10 years ago".
- We not test our parameters on a timespan that is more realistic (as it is not years but months!)

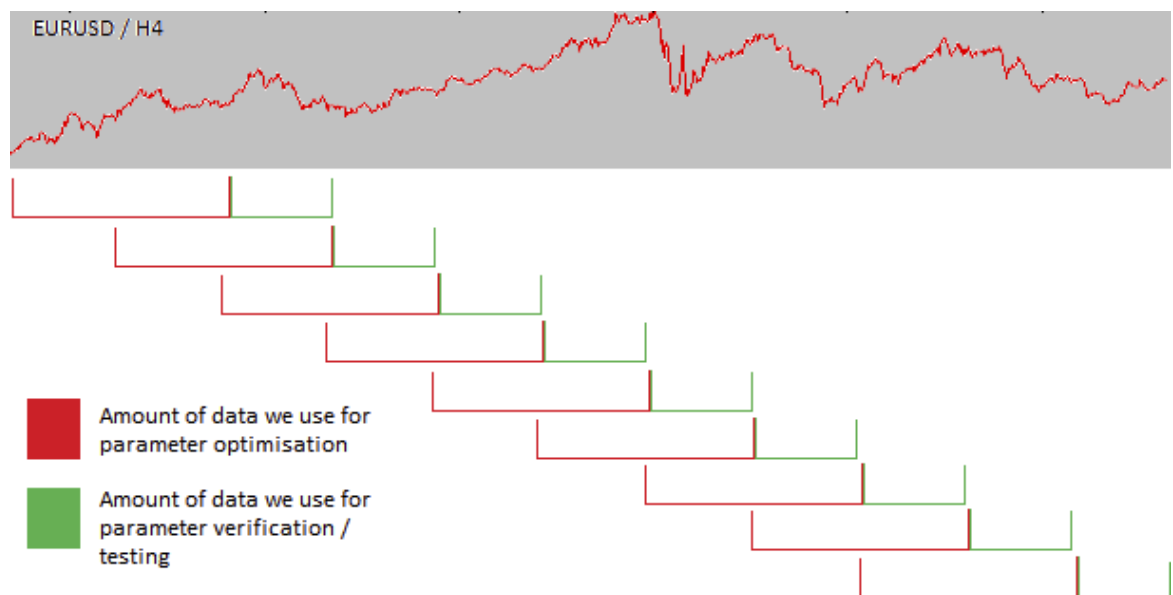
Contra:

- We only used a small part of the available price-data for our tests. This is not very efficient!
- Ok, remember the initial situation, where we have settled on parameter ranges, amount of data to optimise on, and the "desired characteristic". Our analysis has the purpose to verify these choices, whether they are valid or not.

But in this case, we only made one test with them, so we optimised on one part of the data, then we chose 1 parameter-combination and tested it on 1 "unseen" dataset.

Facing the million/billion possible parameter-combinations an EA can have, and the infinite ways the markets can change to generate new and "unseen" behaviour, do you really think that 1 test, 1 datapoint, 1 past->future relationship, is enough to judge from? Of course not!

Step Four: Walk Forward Analysis



So, as you might see, a Walk Forward Analysis is the same thing like doing a normal back- & out of sample-test, but we do it over and over again, so we end up not just with 1 test-case but with many (100-150 in most cases, up to 1000 if we choose very small test-period).

That way we can verify our system + our optimisation-methodology on many, many independent test-cases, which is THE reason why we want to use WFA instead of every other analysis-method described in here.

Pro:

- For our final analysis-report, we only take into account the green test-results, as they are the "unseen future" relative to the red optimisation-windows. That way, we simulate the same process we would face during live trading: Optimisation on the past, trading on the (relative) future!

That allows us to draw meaningful answers to the initial question, as we only analyse performance in "the future".

- We use all data available for our testing
- We have 100-150 independent "PAST=>FUTURE"-relationship-tests, which gives us a clue about the future performance, not the past performance!
- We avoid overfitting, as we use different datasets to optimise and verify our parameters
- If we want to trade live, we simply make "one more step" of the WFA, optimise on the last available data (the "red" dataset would then end at the end of the chart), and then trade "in the future" (the "green" dataset would be our live trading). So we trade the system using the EXACT same methodology we have tested 100-150 times already.
- Due to the frequent re-optimisation of parameters, the EA is also continuously re-adapted to the markets, which will most likely increase the overall profit.
- A traditional backtest answers the question "How good was my EA in the past", whereas a Walk Forward Analysis answers the question "How good will my EA be in the future, during live trading".
- It does not only evaluate an EA, it also evaluates the corresponding trading plan that determines how to pick the best parameters for live trading.

Contra:

- Most EAs will not pass this test. But this is not bad, because let's be honest, almost all EAs in existence are bullshit. So if almost all EAs tested with this approach would give bad results, that would be great.

Even if a lot of people do not like to be disillusioned about their "holy grail money printing machines", it's better to face the truth during EA-development and not during live trading.

"Contra":

- There are some limitations regarding this process which are discussed in the [third article](#)

The main advantage is that we get 100-150 (more or less) independent test-cases, whereas a Backtest+Out-of-sample-test gives us only 1 test-case (or 1 datapoint).

Thinking this one step further, DATFRA can analyse a few hundred thousand (more or less) independent test-cases per system. Learn how and why

Why 'Walk Forward Analysis' is still unreliable and useless!

Well, lets begin. My first concern is that the design of Walk Forward Analysis is, in its nature, unrewarding and not the kind of analysis a trader wants.

Also, I claim that the results of a WFA are more or less random, and if a system works well after a successful WFA, then not because the test was successful, but because the trader designing the system did a good job.

In this article I do not yet want to show how this problems can be solved, I just want to demonstrate that they exist. In my next article I will explain how I think this all can be solved in an elegant way.

The fundamental design problem

Walk Forward Analysis is designed to evaluate a trading construct you give to it.

This construct consists of:

- Trading System (eg an Expert Advisor)
- Market/Timeframe (eg "EURUSD / H4")
- System's Parameter ranges (eg "Moving Average Period from 50-150")
- Optimisation (In Sample) Timespan (eg "Optimise on 2 years of data")
- Forward Trading (Out of Sample) Timespan (eg "Forward Trade for one month")
- Preferred characteristic (eg "forward trade the candidate with highest profit")

So all this has to be pre-determined by the trader, out of intuition, and not based on true facts and data. But god, these are the most important decisions, how should one "guess" them?!

And then, WFA will only be able to tell you if this construct would have worked in the past or not, thats it.

So in order to find the best trading construct, you have to use trial&error and repeat WFA step multiple times. This would then, step by step, even lead to the worst case, your "unseen" out-of-sample tests would slowly become "known" in-sample data and the whole advantage of WFA over backtesting would fade away completely.

This design related problems are already showing that WFA can not be the end of the road in terms of system analysis.

In a perfect world, you should give the analysis algorithms only the trading system and the market/timeframe, no other parameters. And then, the algorithm should tell you the best choices for all the other parts of the trading construct, based on data and facts, not the other way round.

Side Note: it should NOT just tell you how to trade your systems, it should give you the possibilities to look into the system's characteristics on your own. You should never be forced to trust any algorithms without the possibilities to check it's findings!

This is very, very important. It is not very much of value to evaluate a single trading construct, but it is a gamechanger if you can look into your strategies in a way that would allow you to just "see" how they work and what trading construct will work best (More on this in my next article)

Even worse: Unreliable results because of lacking data

Ok, so even if a trader could come up with a good trading construct out of intuition/knowledge, WFA would still be a more or less random thing. But first, let's make a rough calculation:

An example trading system and a small estimation of its parameterspace-size

So, a system that enters trades based on a Moving Average Crossover and RSI Indicator, and exits them using a different Moving Average Crossover has at least 5 Parameters (2x2 for MA-Periods + RSI Threshold). It's 6 if you take into account the StopLoss.

Let's say the "fast" Moving Average Periods can be 10-50 and the "slow" ones 50-250, the RSI threshold can be 1-100 and the StopLoss 50-150 pips (this is no real system, just an example!)

So this system can already be traded in 40*200*40*200*100*100 different ways. That is 640 billion (640.000.000.000), which is quite a huge number.

One might question my exact example strategy, but can not question the millions or billions of possible parameter combinations, even for small systems.

But thankfully, if we take into account that a lot of these parameter-combinations would behave very similar, we do not need to evaluate them all, but we need at least a meaningful sample of it, like a few hundred thousand or a few million.

So, keep this huge amount in mind, even for small systems, because with every new dimension for our optimisation problem's solution space (every new parameter) the amount of possible parameter-combinations grows exponentially.

Walk Forward Analysis - missed data during optimisation

Ok, now lets look at the first step of WFA, and the first problem: Missed data because of inefficient algorithm design and computing time concerns.

During optimisation step of WFA, the algorithm should, in a perfect world, evaluate all 640 billion combinations in order to determine which of them work best. Of course this is not possible, but a "meaningfull" sample (let's say 500.000) would be feasible and _needed_ if we want to look at the "real" picture.

The problem is, due to limitations of WFA algorithms, optimisation has to be done in every single Walk Forward Window.

Let's say we do a WFA on 10 years of data and our Forward Trading Timespan is 2 weeks: That makes 240 Walk Forward Windows. That means 500.000 tested parameter combinations per window would need 120.000.000 single simulations.

And then, remember that WFA relies on a trial&error principle, so you will most likely have to do this a few times.

You see? Evaluating the "real" picture would take very, very long, and therefore most WFA implementations are forced to only evaluate an very much cropped fraction of the actual parameterspace because it is not possible to evaluate the whole parameterspace (or a meaningfull sample of it) in a reasonably small timespan, because optimisation has to be done in every single WF-Window.

This means, WFA most likely does not evaluate 500.000 parameter combinations per window but only 10.000 or 50.000 or something like that. So eventually we already lose like 90% of all data in this step.

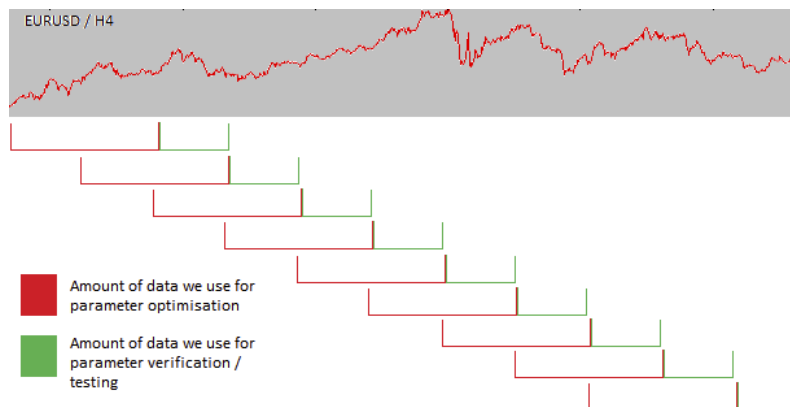
This is a problem that could be solved if the trader has lots of time for his/her analysis (which is not likely, especially based on the trial&error method), or with a more efficient design of these algorithms. Nevertheless, in praxis, this problem is ever-present.

For comparison: DATFRA, which is my private research project, only has to do one single simulation per parameter-combination, no matter how many WF-Windows it analyses. In the above example, that would already decrease the computing time by the factor 240.

Parenthesis: What kind of data do we look at when analysing trading systems, what is a "datapoint"

I will talk about "datapoints" and "data" quite frequently in this article and in my posts, so here is an explanation. When analysing systems, it is always about a trinity of informations.

Remember how WFA works:



So a datapoint, of which 1 is generated per Walk Forward window, consists of:

- The performance in the RED optimisation window
- The performance in the GREEN forward trading window

- The used parameter-combination for this specific test

So, in our example, a WFA would generate 240 of them, whereas 120million (500k * 240) would be possible for our example system. That should already give you headache.

Walk Forward Analysis - tons of missed data during forward trading

Ok, now lets look at the second step of WFA, and the second problem: Missed data because of `_wrong_` algorithm design and computing time concerns.

Now remember, a meaningfull sample of our trading system's parameterspace would be 500.000, and we have 240 WF-Windows. That would make a total of 120.000.000 optimisation-candidates. And out of this huge amount, a WFA algorithm takes the very best per window, 240 in this example.

That is 0,0002% of the total amount of all datapoints that we could use to describe/analyse this system and it's ability to produce good forward trading results, based on good optimisation results.

And then WFA takes these few datapoints and claims it gives a somehow realistic view on a trading system's performance / robustness.

Thats nonsense! You also would not judge a picture's colour by looking at 1 pixel, would you?

A word about fluctuations and why the "very best" parameter combination is not meaningfull

You could argue that it is not important if we forward trade all 500.000 candidates per window, because we are only interested in the top performers, as they are the ones we trade in realtiy.

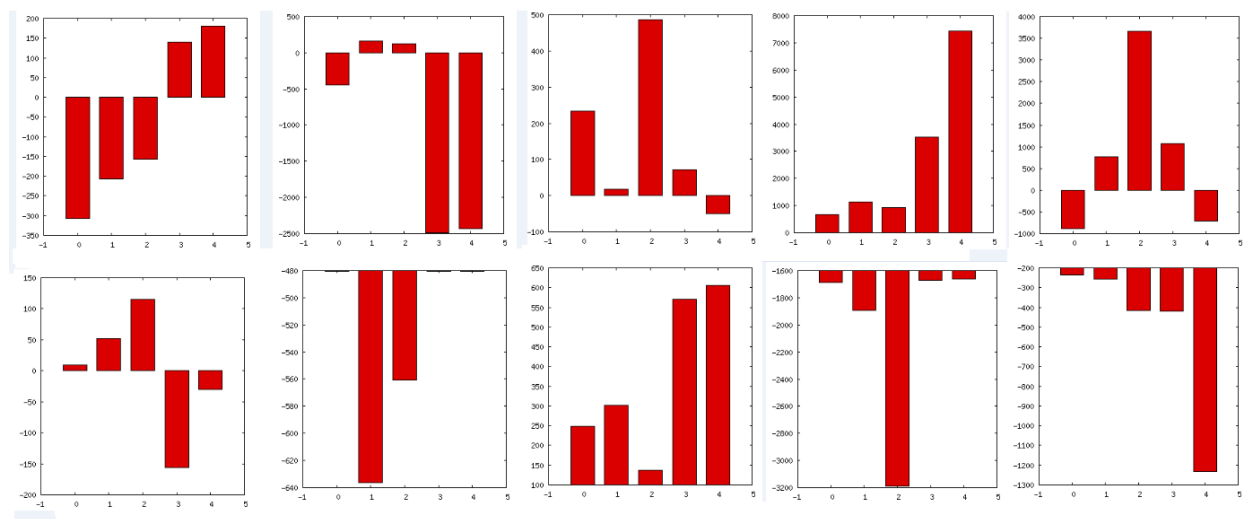
Well this argument would `_only_` works if:

- We would ignore the ~90% of data lost in the optimisation step
- The very best candidates would be meaningfull, which means that all candidates that are following (like the next 10 or 20 or 50, which is not much compared to 500.000) would behave in quite the same way.

But reality is different, the performance of the top candidates per window fluctuates quite much and taking the "very best" therefore leads to more or less random outcomes.

Experiment 1

Here are some examples, I plotted the forward trading performance of the best (left) and the next 4 candidates of some random strategies I created and evaluated with DATFRA. Most of the analysed WF Windows looked like these:



These were just a few examples to illustrate my point of view, I could show hundreds or thousands of them.

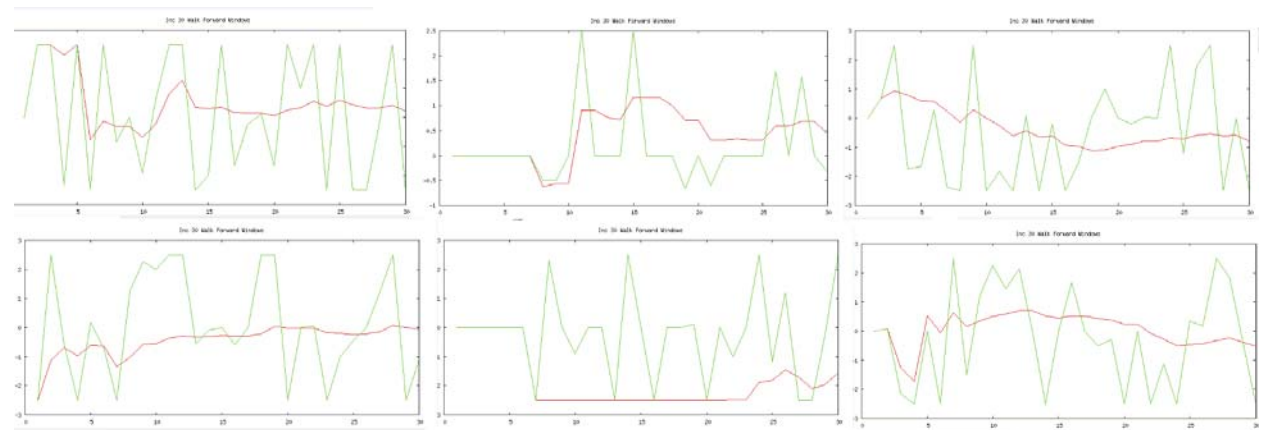
So, for the real picture, you would AT LEAST need to evaluate a few hundred of the top candidates, not just one, as it does not show the "real" picture. It's performance is more or less random!

A perfect analysis algorithm would evaluate every single candidate that made at least 1\$ profit during optimisation. That would give the real picture and most likely 1000 or 10.000 as many datapoints than what a WFA gives.

Experiment 2

Here are some more examples, this time I plotted the overall WFE (red) and the WFE of single windows (green) of some random strategies I created and evaluated with DATFRA.

WFE (Walk Forward Efficiency) is a measurement that compares in-sample and out-of-sample performance and is used as THE statistic about system robustness in WFA (google for it if you want to know more about it)



This clearly shows the fluctuating nature of the results a WFA generates, and that the end result is not really telling much about your expected live trading performance.

Btw: To keep the plot scale in limits I did map all points > 2.5 to 2.5 and all points < -2.5 to -2.5, so reality is even a lot worse. That is also the reason why the second image in the second row does not look "right"

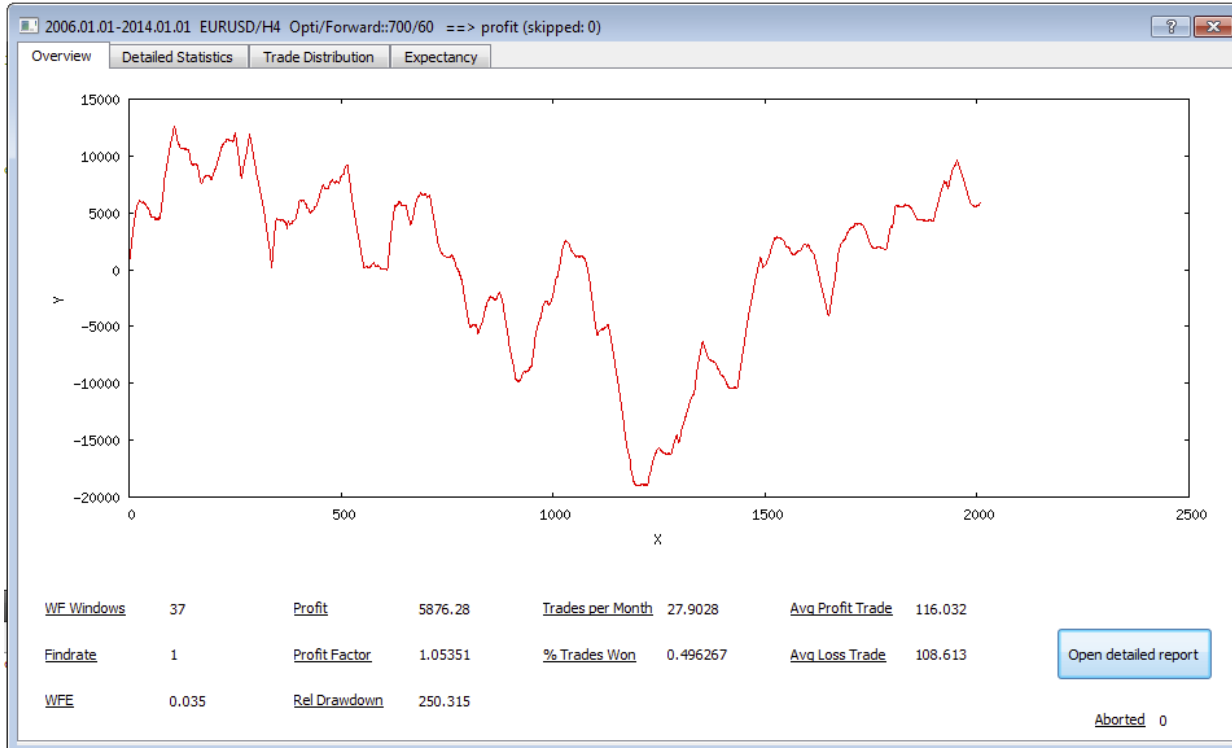
Experiment 3

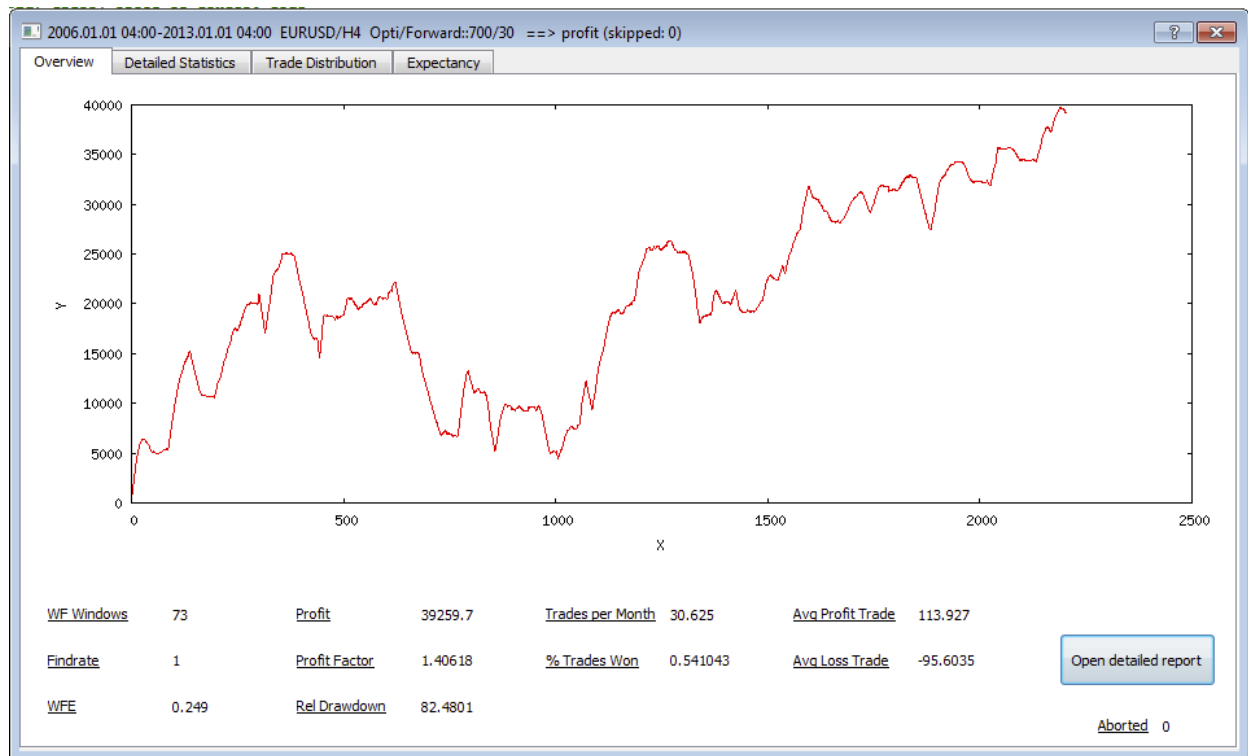
I did repeat the same Walk Forward Analysis multiple times. I used the exact same parameters each time: EURUSD H4, optimisation was done using MT4's genetic algorithms, Optimise on 700 days, trade for 60 days.

The tested system was very simple: It uses 2 Moving Averages for entry, 2 for exit and has 6 parameters total (4xMovingAverage period, 1 "difference in pips", 1 stop loss).

And here are the results of all 5 tests, I guess I do not have to comment them.







A word about feasibility

Please do not think I only talk about grey theory here "as it is not possible to do this kind of simulations in a short enough amount of time anyway".

If the algorithm is designed well, one would not need a single further simulation in order to determine forward trading profit and not a new optimisation procedure for each WF-window.

So for the used example, DATFRA can generate 34.000.000 "Optimisation=>Forward Trade datapoints" in ~24 hours and on a mid-end PC (8GB Ram, quadcore 3GHZ).

Still not 120millions, sure, but compared to 240, I think its a very good result.

So it IS feasible to analyse a system with such a level of insight, even on today's hardware.

Parameterspace Analysis - DATFRA's way of analysing systems

This is quite much to read, true, but I promise that you will find the article very usefull and that the concept layed down here is superior to every other algo trading approach you have ever heard about. :)

Btw: Please read my other 3 articles before you continue with this one, you can find them here on the forum.

How Parameterspace Analysis (PSA) works, Part 1 - What is the fundament of all, what data do we work with?

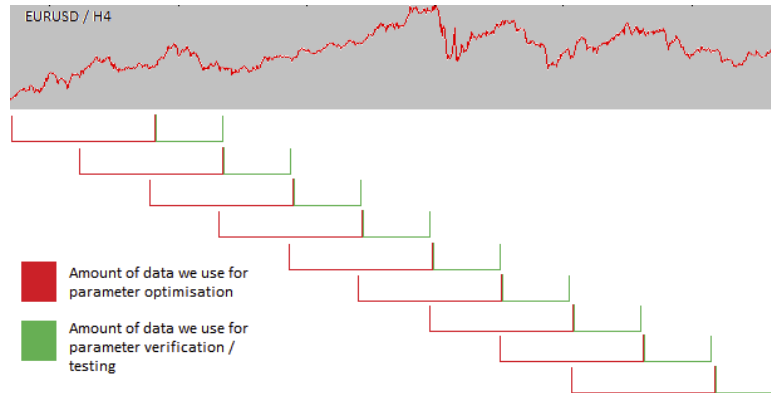
Parameterspace Analysis is no fixed and inflexible method, it is a datamining approach that first collects all important data of a system, and then gives you the tools and interfaces to analyse and dig into this data.

It works with all Metatrader4 Expert Advisors that trade only on bar opening - otherwise generating the huge amount of data would not be possible.

What kind of data do we look at when analysing trading systems, what is a "datapoint"

First of all, I will talk about data and datapoints frequently. So what is a datapoint?

Let's first recall how Walk Forward Analysis (and also Parameterspace analysis works):



One datapoint can be generated PER PARAMETER-COMBINATION(CANDIDATE) and PER WINDOW. It consists of:

- The performance in the RED optimisation window
- The performance in the GREEN forward trading window
- The used parameter-combination for this specific test
- (Some statistics about the RED window as a whole. For example, how many candidates were profitable in optimisation, what was the average profit, what the average profit factor etc)

So in WFA, one datapoint per window is generated, as it only forward trades the very best candidate per optimisation. In PSA, one datapoint per candidate is created.

For comparison, a Walk Forward Analysis usually works on ~50-250 datapoints. So it does 50-250 optimisations, takes the "best" result, forward trades them -> 50-250 datapoints.

PSA, in contrast does... well, read on.

What is the actual difference between WFA and PSA?

PSA basically does the same thing that WFA does (look at the image above), but after each red frame (optimisation), it takes all parameter-combinations (==candidates) and forward-tests them, instead of only the "best" one.

(Btw: One can choose if it should forward trade them all or just the candidates that made at least 1\$ profit)

Then it saves each of these "optimisation->forward trading" datapoints to a database (this is what I call the **parameterspace of a system**).

Compared to a WFA this can easily produce 1.000 or 10.000 as many datapoints, that describe a given trading system.

And there lies the power of PSA, algo trading is about data and information, and it makes sure you get as many data and information about your system as somehow possible, so you can make funded decisions.

What might now just sound like "the same thing" (but with more datapoints) leads to a variety of possibilities that are simply not possible with common WFA.

PSA by example - and some ideas how this data can be used

Taking my old example strategy (from article about Walk Forward Analysis) "If the price moved more than X pips in the last Y days, a course correction will happen", such a parameterspace-database could look like that (its a simplified example of such a parameterspace, where each row is one datapoint):

Optimised on: 01.Jan.2001 - 01.Jan.2003		Forward Traded on: 01.Jan.2003 - 01.Jan.2004		
STRATEGY-CANDIDATE		FITNESS DURING OPTIMISATION		Profit in forward trading / unseen market conditions
If the price moved more than 100 pips in the last 2 days, a course correction will happen		100\$ profit	1.2 profit factor 10% drawdown	70\$
If the price moved more than 150 pips in the last 2 days, a course correction will happen		100\$ profit	1.3 profit factor 9% drawdown	75\$
If the price moved more than 200 pips in the last 2 days, a course correction will happen		175\$ profit	1.4 profit factor 7% drawdown	85\$
If the price moved more than 100 pips in the last 3 days, a course correction will happen		110\$ profit	1.2 profit factor 10% drawdown	77\$
If the price moved more than 150 pips in the last 3 days, a course correction will happen		170\$ profit	1.4 profit factor 8% drawdown	86\$
If the price moved more than 200 pips in the last 3 days, a course correction will happen		190\$ profit	1.5 profit factor 7% drawdown	90\$
Optimised on: 01.Jan.2002 - 01.Jan.2004		Forward Traded on: 01.Jan.2004 - 01.Jan.2005		• • •
Optimised on: 01.Jan.2003 - 01.Jan.2005		Forward Traded on: 01.Jan.2005 - 01.Jan.2006		• • •

Rank based evaluation - normalising absolute values

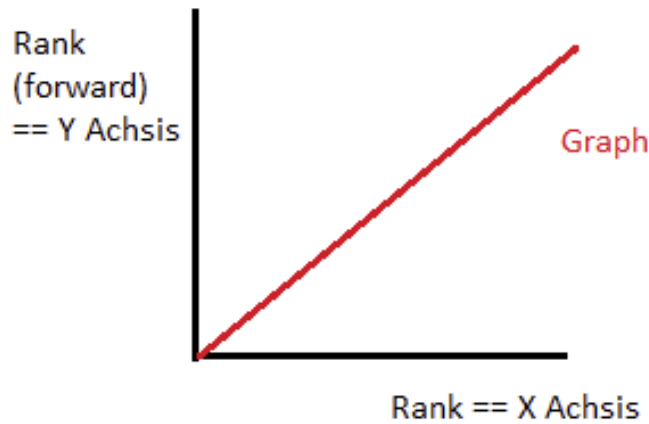
If we analyse absolute fitness values, like profit, we do not take into account that sometimes a strategy works better, sometimes it does not. Also we do not take into account different optimisation and forward trading timespans.

So in order to have something "normalised", I am using ranked fitness values for some analysis methods. (Correlation analysis in particular). These ranks express the fitness relative to all other candidates in one window!

An example should make this more clear, here we see the "profit rank", and how it is calculated from "profit":



So, for a perfectly robust system the rank during optimisation and the rank during forward trading, would always be the same. Plotted, it would look like this:



So, based on how much our actual graph is different from this optimum, we can simply see how robust our strategy is. No wired statistics, no unproven theories, just the robustness in one picture!

Some ideas on how this data can be used

Make sure you fully understand how this PSA-Database is structured, it is essential!

Forward-trading all candidates instead of the "best one" is just the logical next step from WFA, but this approach gives us enormous advantages:

1. No data is thrown away, it is all saved to make sure our evaluations are highly reliable and robust!

2. No assumptions have to be made. In WFA, you have to decide some things, before you start the analysis. For example, what optimisation fitness value (like "highest profit" or "lowest drawdown" etc) is used to pick the candidate, from optimisation, for live trading. So some very important decisions have to be made without solid data, based on intuition, which is very bad, of course.

3. All the different parameter combinations, over all the different "past->future" windows, are in the database, so we can just see which parameters are best for our system, no educated guesses or trial&error needed. Perhaps a moving-average of 100-200 is best? Or is 50-100 better? Who knows...

4. Correlation tests based on ranks: As we have all the "past->future" datapoints in the PSA, we can see how robust our strategy performs, how high the chance is that we get good live trades out of good backtests/optimisation on past data.

5. We can see what market conditions our strategy is suited for. For example we can spot simple thresholds for it, like "if a strategy makes >XYZ\$ during optimisation, it has a very high chance to make a good profit in forward trading." Same thing for overfitting, we could see when our strategies tend to be overfitted, with the same threshold approach.

6. I am currently experimenting with this, but it shows very, very promising results: We can train artificial intelligence (in my case Artificial Neural Networks) on all this data. And the AI can then tell us if we should trade a given strategy at the moment, with what risk we should trade it and how likely it is that it will make profit in the next weeks or days :)

All these evaluations are not possible with a WFA, as it is not "producing" the data needed for it. And these are just the first ideas, I am sure the community will come up with dozend of ways to analyse this data treasure

How Parameterspace Analysis (PSA) works, Part 2 - Practical examples: How PSA data can be used to analyse a trading system

Now that you understand how PSA works and what data it generates, and what could be done with it from a theoretical point of view, let me show you DATFRA as an example, and what power comes with all this information.

A word about the analysed trading system

All examples were done with the default moving average expert advisor that comes with every mt4 installation. That is because I did not have time to develop strategies yet, as I am still busy coding all this, so do not expect the examples to show any good results!

A word about findings and testing

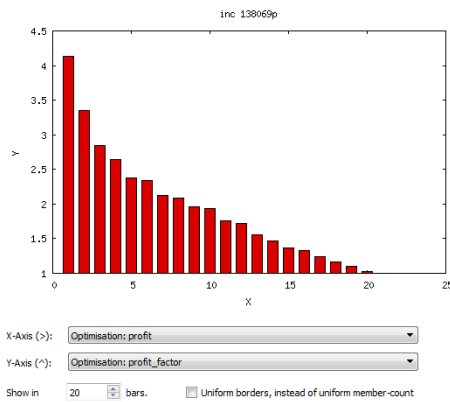
First of all, as all the data we could need already is in the PSA database, we do never have to re-do any simulations. So if we find that parameter X works best if it is between 75 and 100, we can simply filter our PSA database, so it just contains datapoints with where "parameter X" is [75,100] and then continue our analysis.

Or if we want to look into the characteristics of all candidates that were making very much profit during optimisation, we could filter the database to only contain these candidates.

Example 1: How are different Optimisation Results related

If we want to know if higher profits during optimisation are due to higher trade frequency, or higher profit factor or whatever, we can simply look at the corresponding data.

For example, this strategy achieves higher profits due to higher profit factor

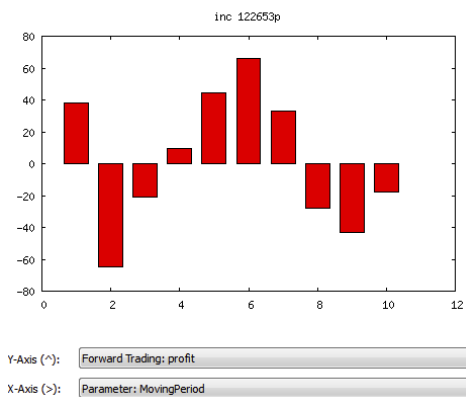


	Optimisation: profit	Optimisation: profit_factor	Cnt
1	10230.9 - 3687.39	4.13887	6901
2	3687.39 - 3003.51	3.35188	6904
3	3003.51 - 2486.24	2.84215	6906
4	2486.24 - 2176.41	2.64205	6903
5	2176.41 - 1873.78	2.30332	6903
6	1873.78 - 1642.58	2.33639	6905
7	1642.58 - 1437.26	2.12073	6880
8	1437.26 - 1260.61	2.09075	6927
9	1260.61 - 1110.69	1.89533	6904
10	1110.69 - 984.29	1.93459	6902
11	984.29 - 863.66	1.76058	6905
12	863.66 - 748.21	1.72803	6900
13	748.21 - 622.6	1.65235	6900

Update Distribution

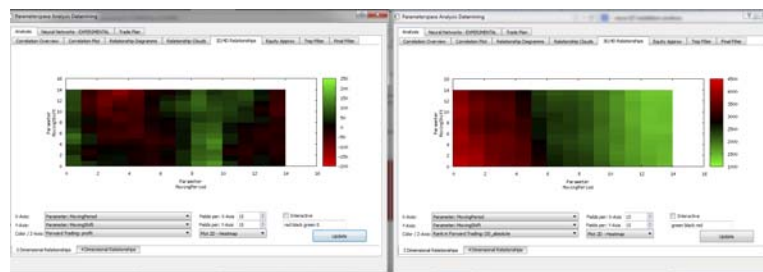
Example 2: What parameters work best

A simple plot of forward trading profit against a parameter (like Moving Period") shows you what "Moving Period" works best: in this case 60-140!



	Parameter: MovingPeriod	Forward Trading: profit
1	3.002 - 22.7017	37.8564
2	22.7017 - 42.4014	-64.7513
3	42.4014 - 62.1011	-21.2399
4	62.1011 - 81.8008	9.68725
5	81.8008 - 101.5	44.7244
6	101.5 - 121.2	65.8609
7	121.2 - 140.9	32.9246
8	140.9 - 160.6	-27.723
9	160.6 - 180.299	-43.1334
10	180.299 - 199.999	-17.5525

But that is not all, you can also analyse 2 parameters at once, in relation to their forward trading profit and/or forward trading drawdown:

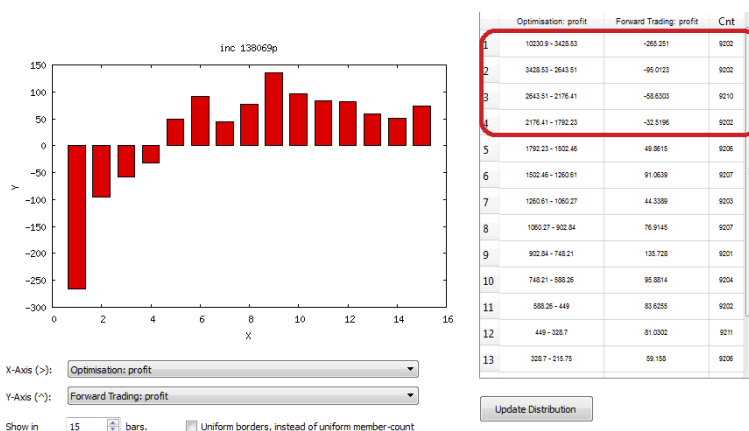


You see how simple it can be? The above heatmaps clearly show that "Moving Shift" parameter is more

or less irrelevant and that a "Moving Period" parameter around the 9th field would give best Profit and Drawdown during live trading!

Example 3: Spot overfitted solutions

Here is an example that shows how easy it is to know when our trading system is overfitted:

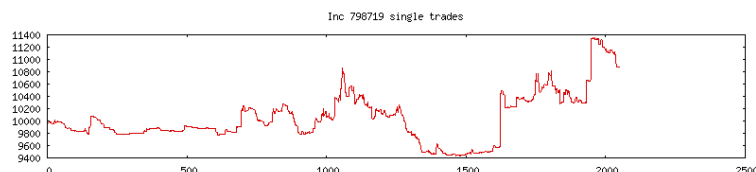


As soon as the profit during optimisation is >1800\$, profitability in live trading is gone and the system tends to lose money - yes, it can be as simple as that to know when a system is overfitted.

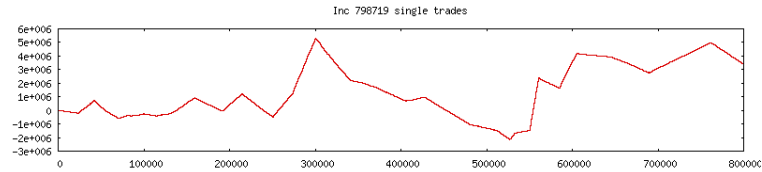
Example 4: You like equity curves? No problem!

What might look like a backtest is actually much more: It consists of ~800.000 single trades!

This example consists of every single entry in the PSA, by simply averaging the single trades on a per-day basis you get a nice equity:



Second possibility is not to average all trades on a per-day basis, but simply to sum them up:



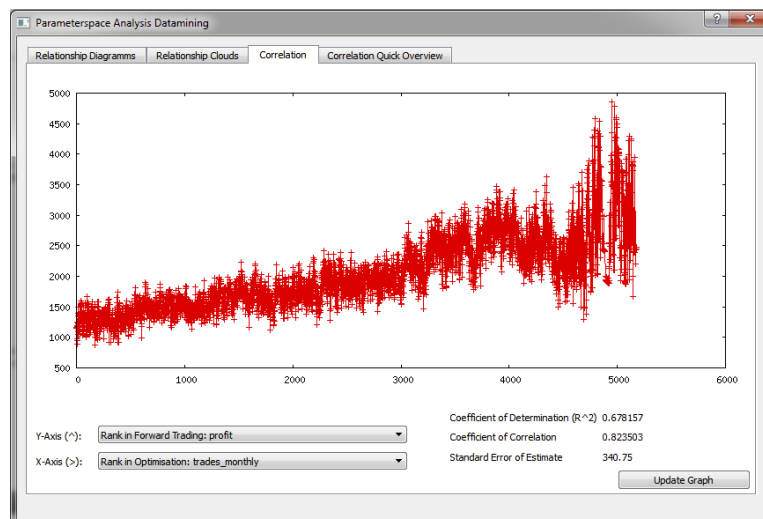
Example 5.1: Evaluate rank based relationships / correlation between parameters - visual

Here we can see the direct correlation of "Trades per month in optimisation" and "Profit in forward trading".

That way, we can analyse how meaningful a given optimisation-fitness value is in regards to forward trading profit.

Remember the 45degree "perfect curve" from 'Rank based evaluation - normalising absolute values'?

Well, here is a real example:



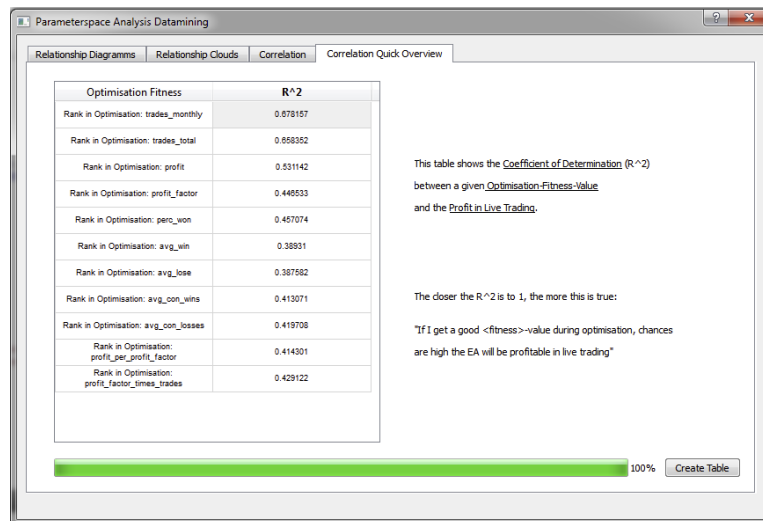
Example 5.1: Evaluate rank based relationships / correlation between parameters - statistical

Remember the 45degree "perfect curve" from 'Rank based evaluation - normalising absolute values'?

We can measure how much our actual correlation plot differs from it with the "Coefficient of determination" (R^2). That is simply a number between 0 and 1, where 1 means perfect correlation.

So if we analyse rank-based correlation between every optimisation fitness value and forward trading performance, we can very easily spot how to pick the most promising candidate for forward trading!

Here is an example:



Example 6: Simulation of multiple WFAs, within seconds!

Remember when I wrote, in my last article, that WFA is bad because it always picks the top result? Well, having all the data already in our database, we can simulate WFA results, but with randomisation in "best candidate choosing":

But of course this is not just an ordinary WFA, it is a way to verify, test and simulate everything that you have found during parameterspace analysis.

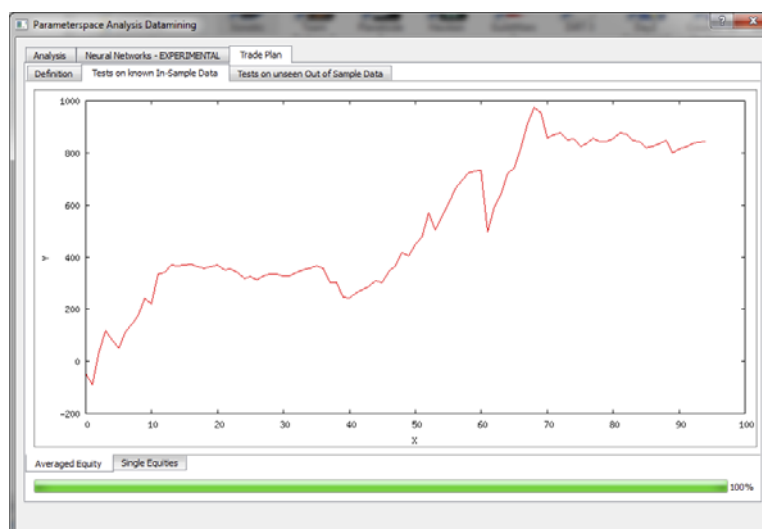
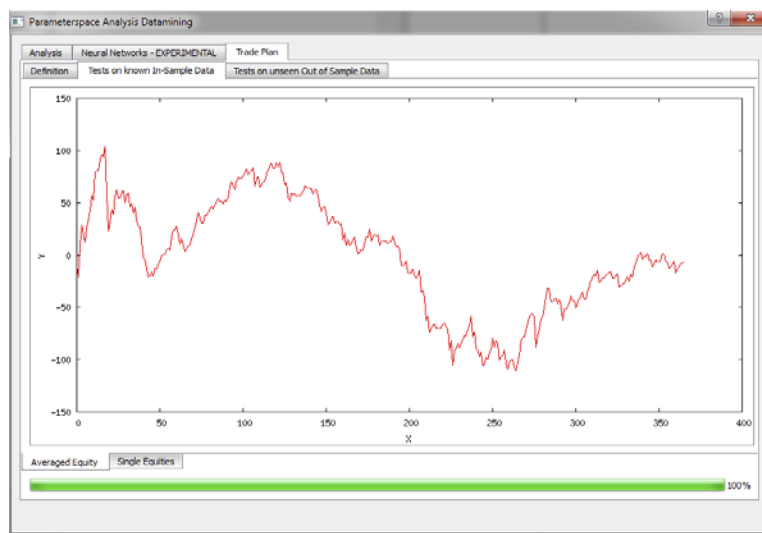


Example 7: A small demonstration of the power of AI backed trading and risk management

Well, this is still research territory and it will take time until it works perfectly, but here is a small peak.

The first equity is an averaged equity of multiple WFAs - of the original strategy

Second equity is the same thing - but this time I use a Neural Network in order to predict what trades to take and what trades are too dangerous:



"Does this really work?"

"Does this really work, does it make profit?" is one of the most common questions regarding everything in trading.

But by now, it should be obvious that the question is not a valid one, as DATFRA, in its core, just mines for and processes data. It simply gives you more insight into your strategy, a higher grade of understanding, and is not meant to be a holy grail money printing machine. (Tough I am working on this part haha)

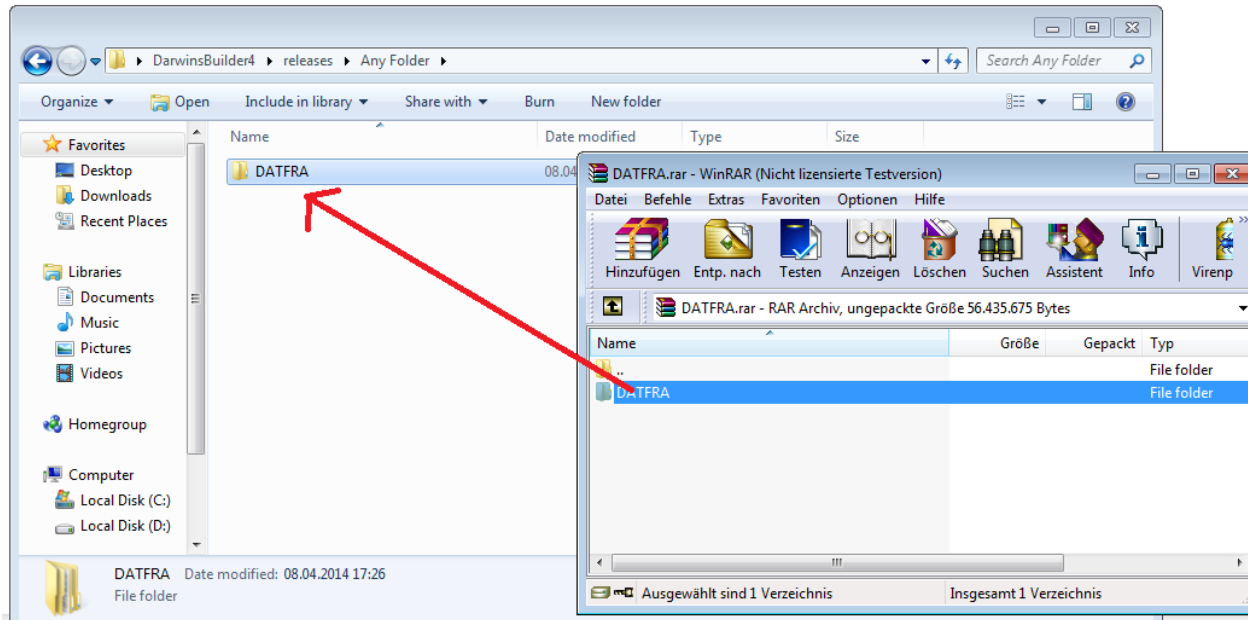
If one believes in the advantages a Walk Forward Analysis brings, it should be clear that this is the same concept, just one step further. If one does not believe in WFA, then there is still much to be learned about algo trading in general ;)

So, there can be no "profit based" evaluation of this method, as it always depends on the trader that uses it (as its just a tool, and a very flexible one) and the strategy analysed with it.

Remember: This is just a framework, a tool. No strategy and no fixed method. You can also not ask as if "Metatrader4 makes profit" or if "Ninjatrader works", because of the same reasons.

Nevertheless I will develop and live trade a few strategies using DATFRA as soon as the alpha version is finished and polished and I have time for this kind of fun.

1. Extract the whole contents of the .rar file to a directory where you have full access to (like Desktop).
NOT TO C:\ OR ANYWHERE ELSE THAT IS PROTECTED AND ONLY ACCESSIBLE FOR ADMINISTRATOR!
Also, there must be NO WHITESPACES in the directory path. Thats nothing I can fix, its because
'powershell' is fucked up and wont find files with whitespaces in the path... dont ask me why



2. Download and install gnuplot FROM HERE



3. Download the latest metatrader4 installer directly from metaquotes (don't use whatever your broker gave you, download FROM HERE). Tough, it should also work with the MT4 version from your broker.



The banner features a blue background with a grid pattern. On the left, there is an illustration of a person with brown hair wearing a green suit, standing in front of three overlapping computer monitors displaying various financial charts. To the right of the illustration, the text 'Trade with confidence' is written in a large, white, sans-serif font. Below this, in a smaller white font, is the text: 'Get the ultimate solution for online trading, technical analysis and automated trading. With MetaTrader 4, you get everything you need to boost your trading.' At the bottom right of the banner is an orange button with a white downward-pointing arrow icon and the text 'Free download (455 Kb)'.

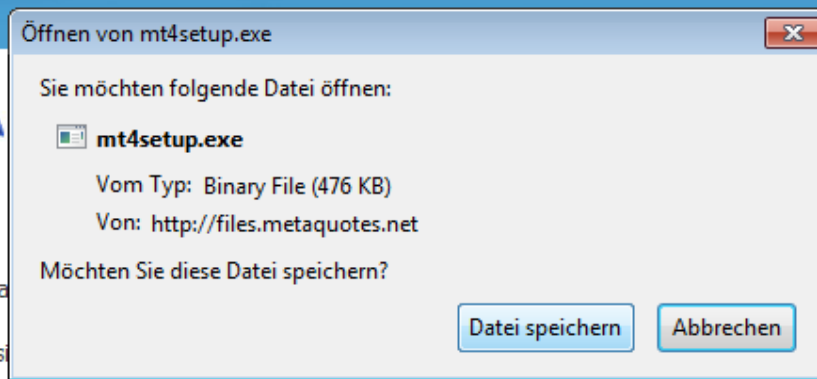
Trade with confidence

Get the ultimate solution for online trading, technical analysis and automated trading. With MetaTrader 4, you get everything you need to boost your trading.

Free download (455 Kb)



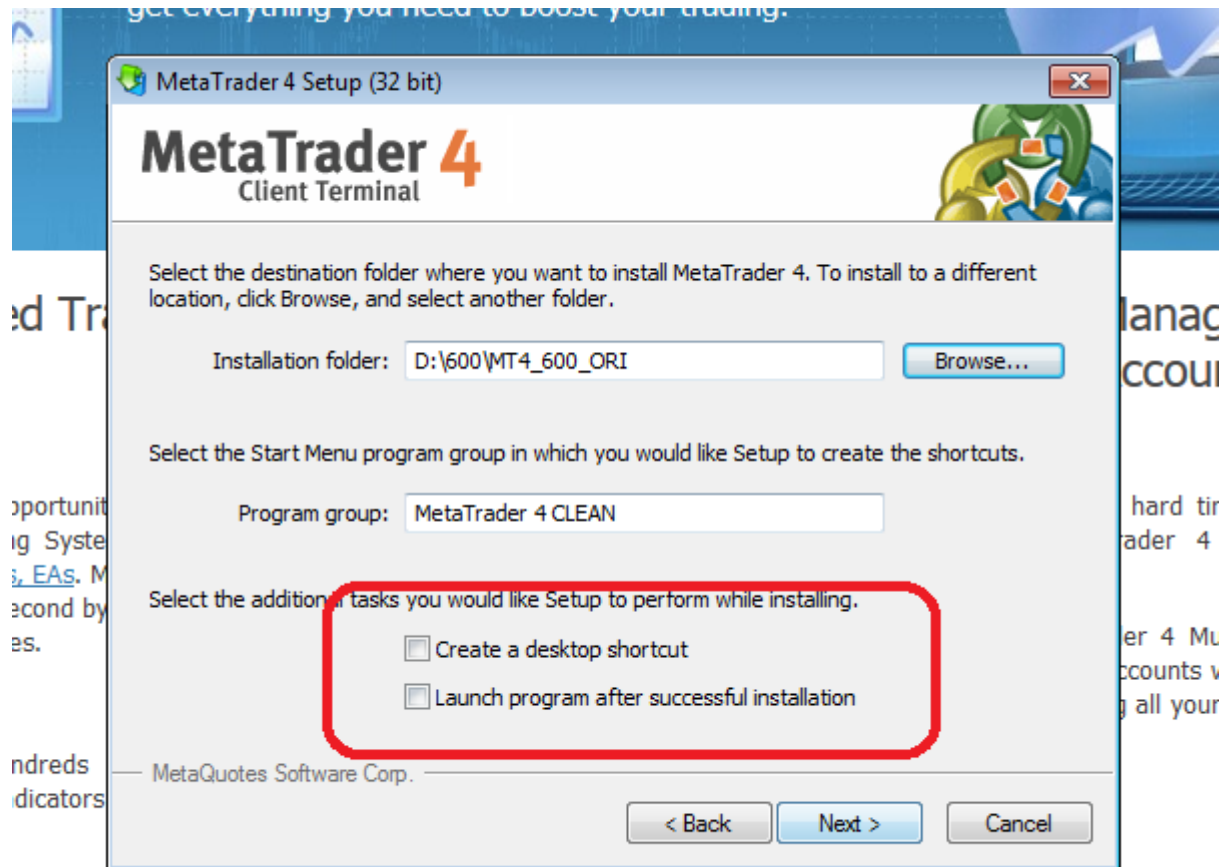
You can increase
MetaTrader 4
trade faster using
can monitor the market

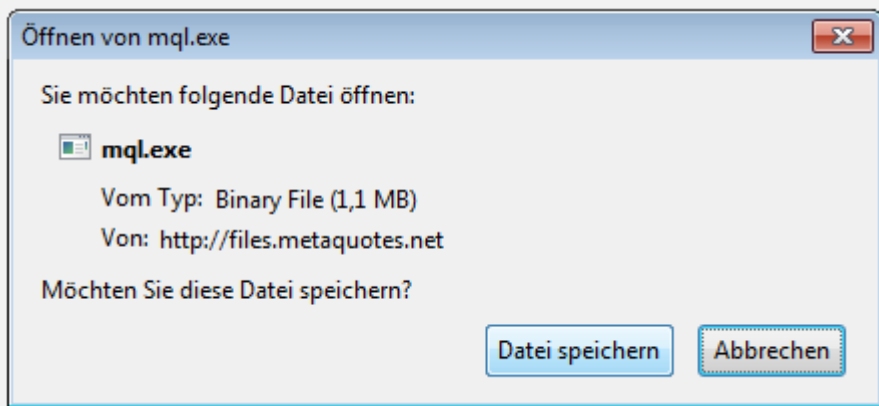


ding

your trading
ccess financial
stay up-to-date
being tied to

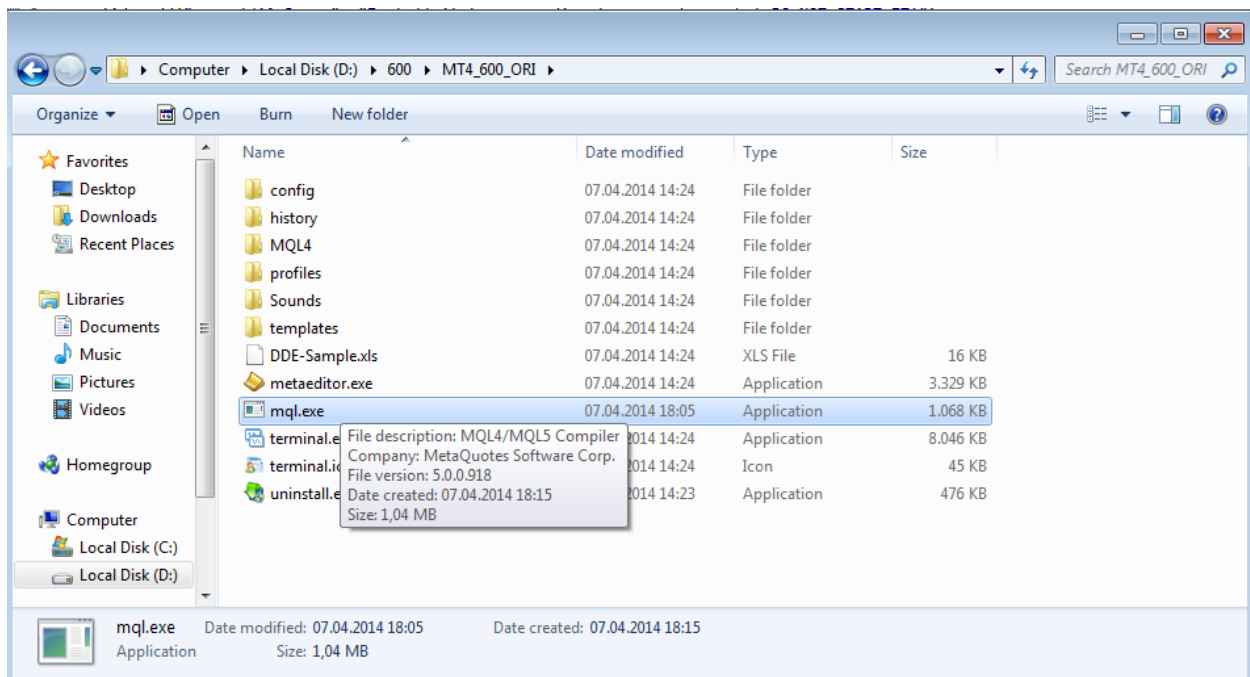
4. Install it to a new directory anywhere, but DO NOT START IT!



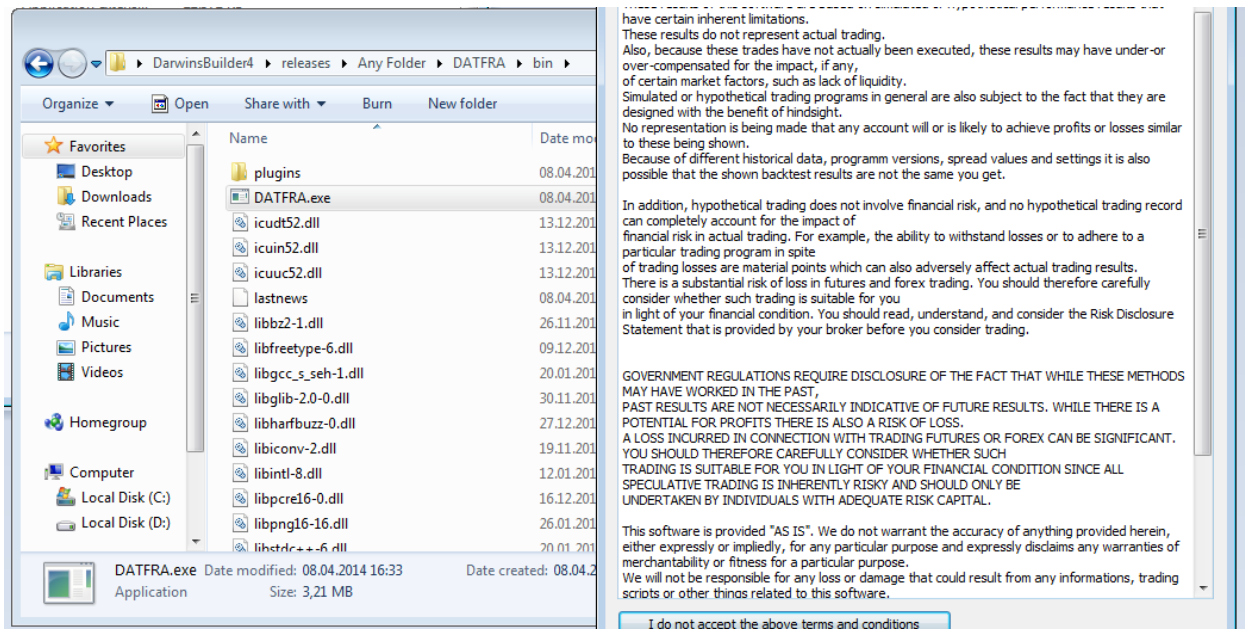


5. Download the MQL Compiler FROM HERE, because it is no longer shipped with Metatrader, and move it to the newly created Metatrader4 installation

6. This is how your MT4 directory should now look like

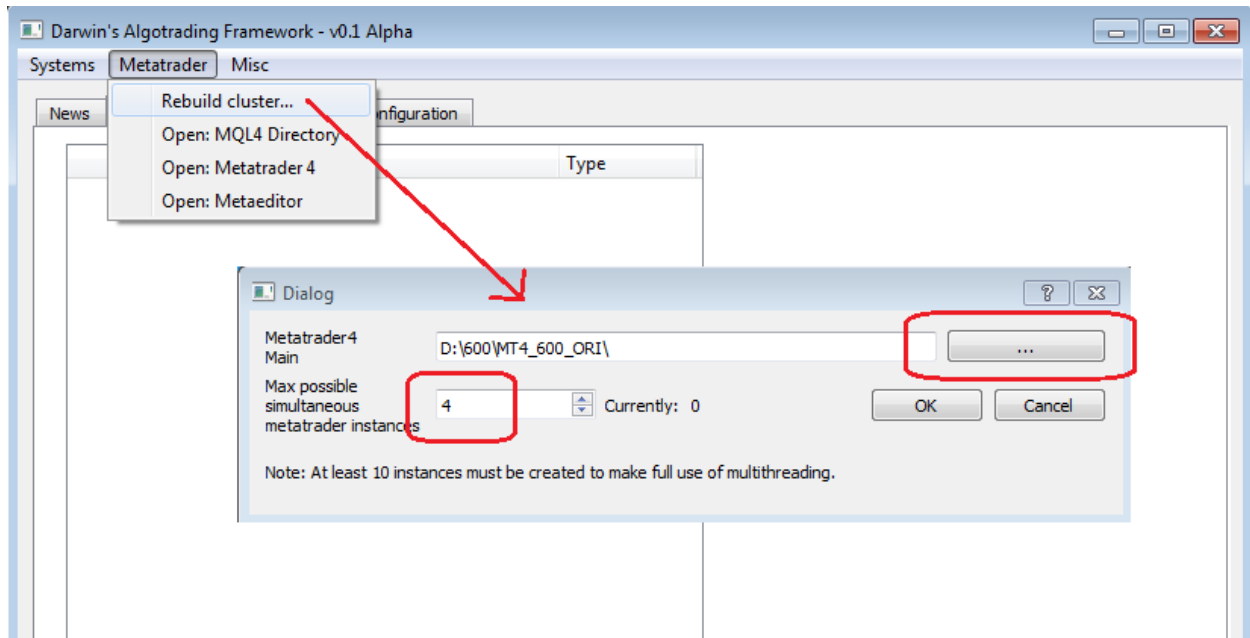


7. Go to the DATFRA directory, go into 'bin' directory and start DATFRA.exe, then accept the terms and conditions. After that, DATFRA will close, so just open it again



8. DATFRA will tell you to rebuild your Metatrader4 cluster. For this, click on 'Metatrader'-'>'Rebuild Cluster'

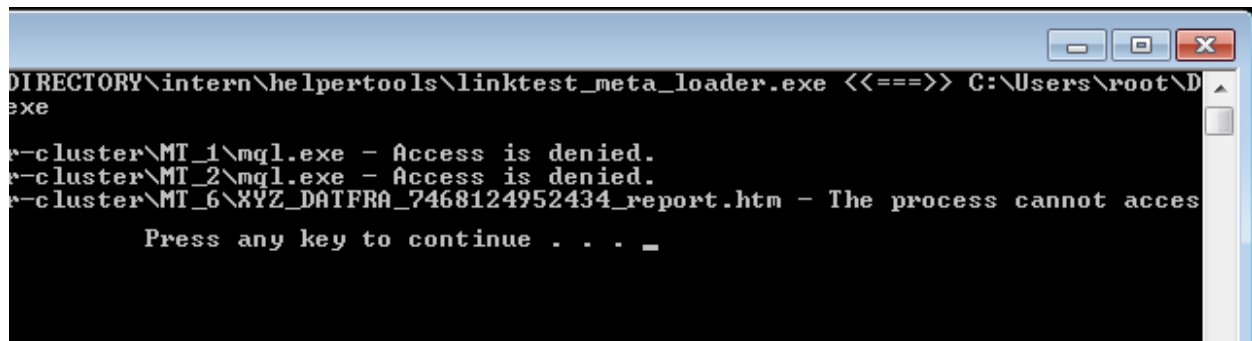
Then click '...' button, select your newly installed Metatrader4 directory, and choose on how many Instances you want to allow (recommended: 10)



9. It will first ask for administrator privileges, then a black command prompt will pop up. When it asks you to, 'press any key', do so and wait for it to finish!

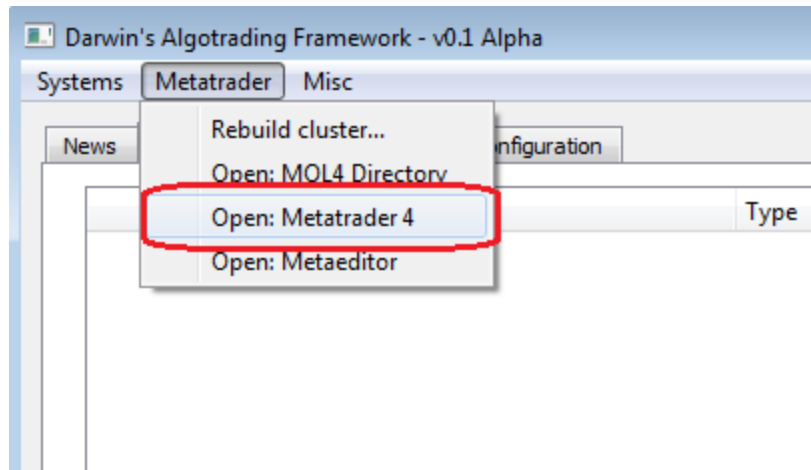
```
C:\Users\root\Desktop\DarwinsBuilder4\releases\AnyFolder\DATFRA\bin\...\n
C:\Users\root\Desktop\DarwinsBuilder4\releases\AnyFolder\DATFRA\bin\...\n
C:\Users\root\Desktop\DarwinsBuilder4\releases\AnyFolder\DATFRA\bin\...\n
C:\Users\root\Desktop\DarwinsBuilder4\releases\AnyFolder\DATFRA\bin\...\n
C:\Users\root\Desktop\DarwinsBuilder4\releases\AnyFolder\DATFRA\bin\...\n
C:\Users\root\Desktop\DarwinsBuilder4\releases\AnyFolder\DATFRA\bin\...\n
C:\Users\root\Desktop\DarwinsBuilder4\releases\AnyFolder\DATFRA\bin\...\n
C:\Users\root\Desktop\DarwinsBuilder4\releases\AnyFolder\DATFRA\bin\...\n
C:\Users\root\Desktop\DarwinsBuilder4\releases\AnyFolder\DATFRA\bin\...\n
C:\Users\root\Desktop\DarwinsBuilder4\releases\AnyFolder\DATFRA\bin\...\n
12 File(s) copied
C:\Users\root\Desktop\DarwinsBuilder4\releases\AnyFolder\DATFRA\bin\...\n
1 File(s) copied
C:\Users\root\Desktop\DarwinsBuilder4\releases\AnyFolder\DATFRA\bin\...\n
1 File(s) copied
Press any key to continue . . .
```

10. ATTENTION: If cluster rebuilding shows you something like that: close the black command prompt, shut down all programs that caused the errors, and then build the cluster. Otherwise it will be incomplete.

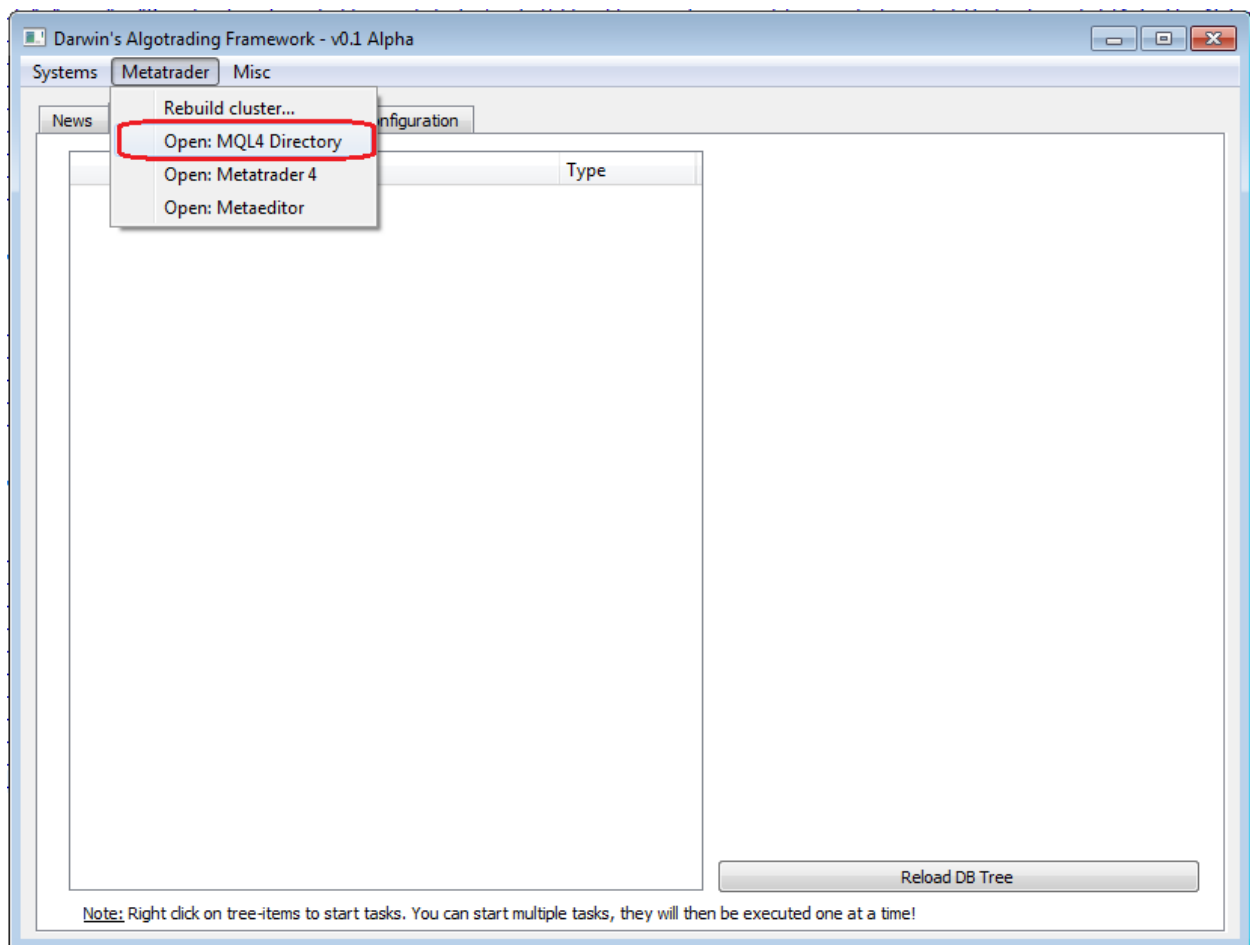


```
DIRECTORY\intern\helpertools\linktest_meta_loader.exe <<==>> C:\Users\root\DI
exe
e-cluster\MT_1\mq1.exe - Access is denied.
e-cluster\MT_2\mq1.exe - Access is denied.
e-cluster\MT_6\XYZ_DATAFRA_7468124952434_report.htm - The process cannot acces
Press any key to continue . . . _
```

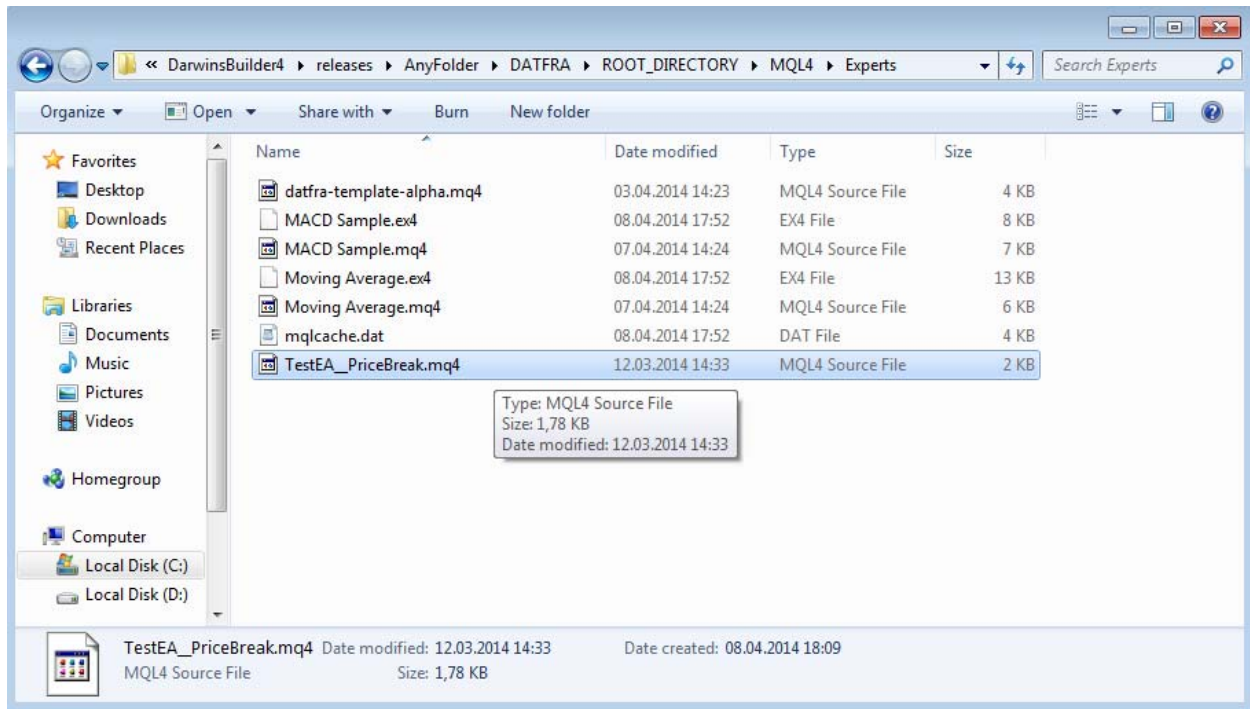
11. To check if everything worked, go to 'Metatrader'-'>'Open: Metatrader 4'. Then a MT4 Terminal should open (it won't pop to foreground, it will stay in the task bar)



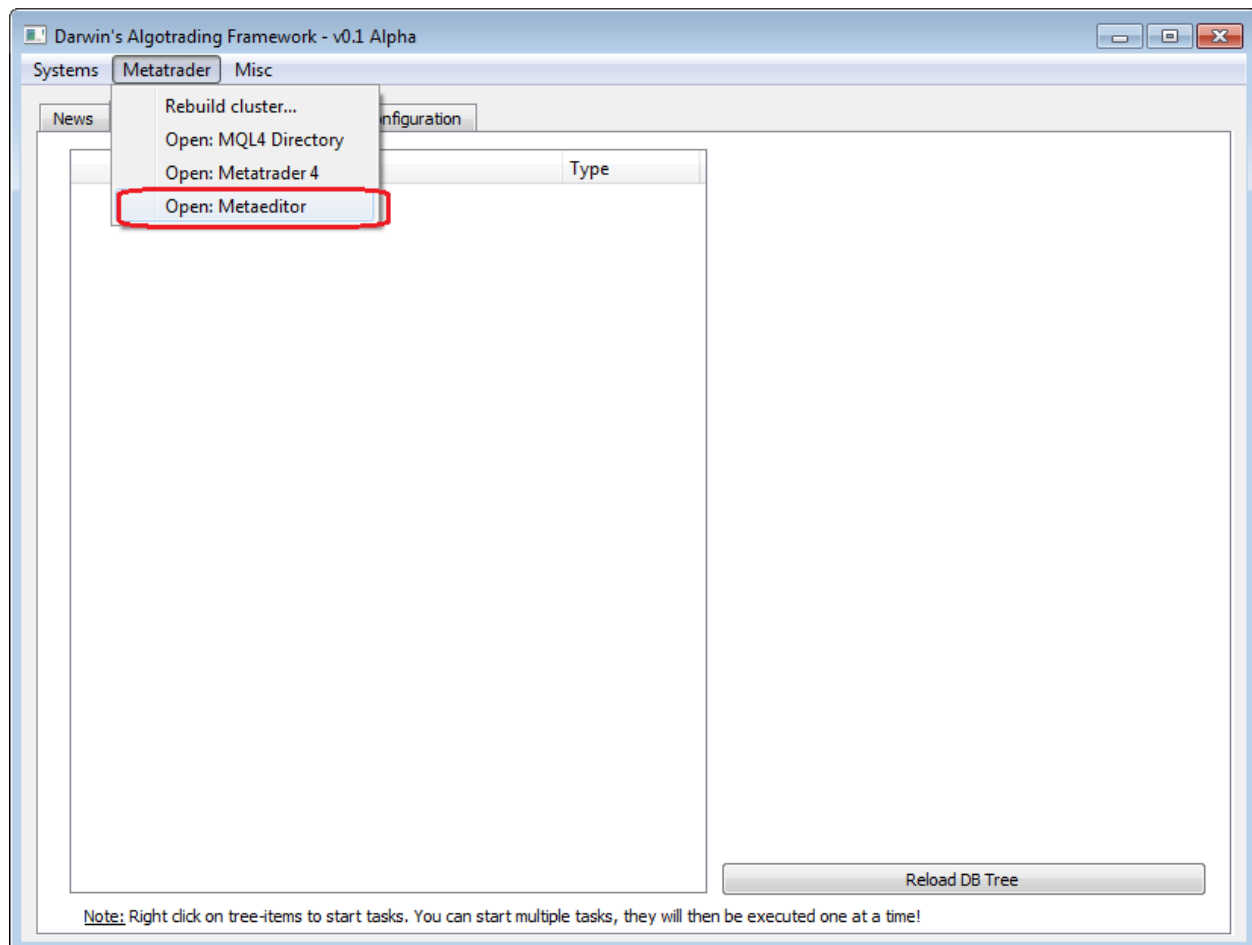
1. Open the MQL4 Directory of DATFRA



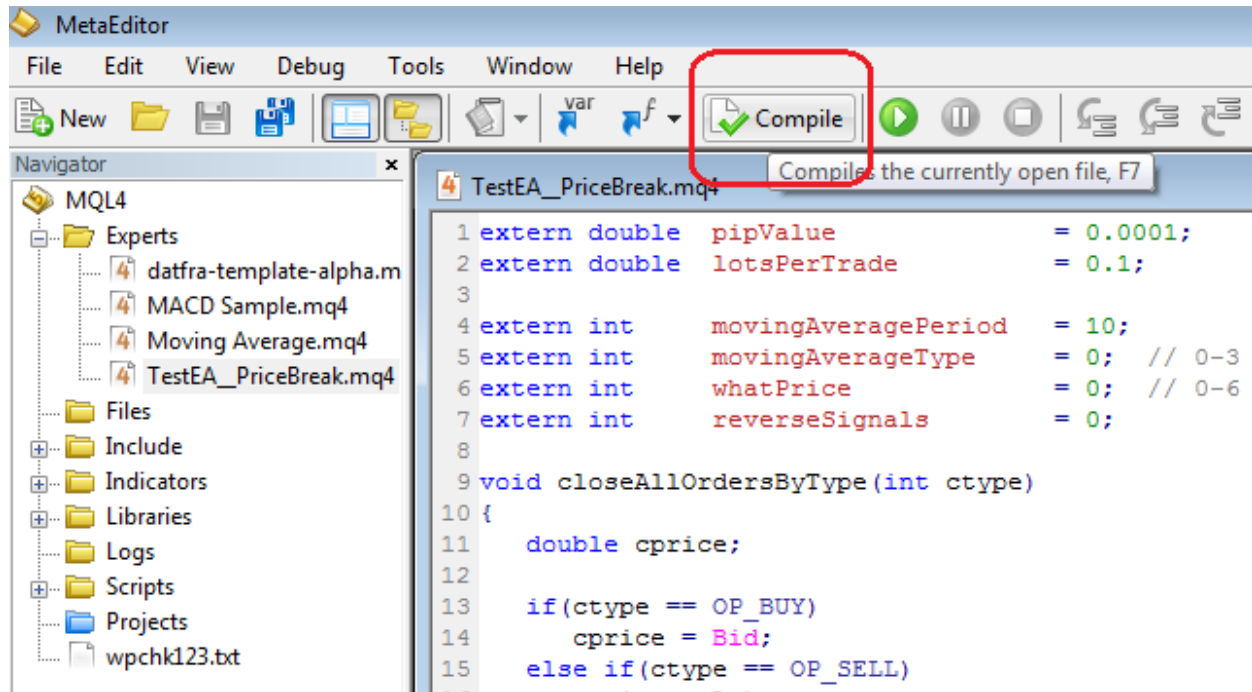
2. Copy whatever EAs you want to analyse, including all include files and indicators, to DATFRA's MQL4 directory!



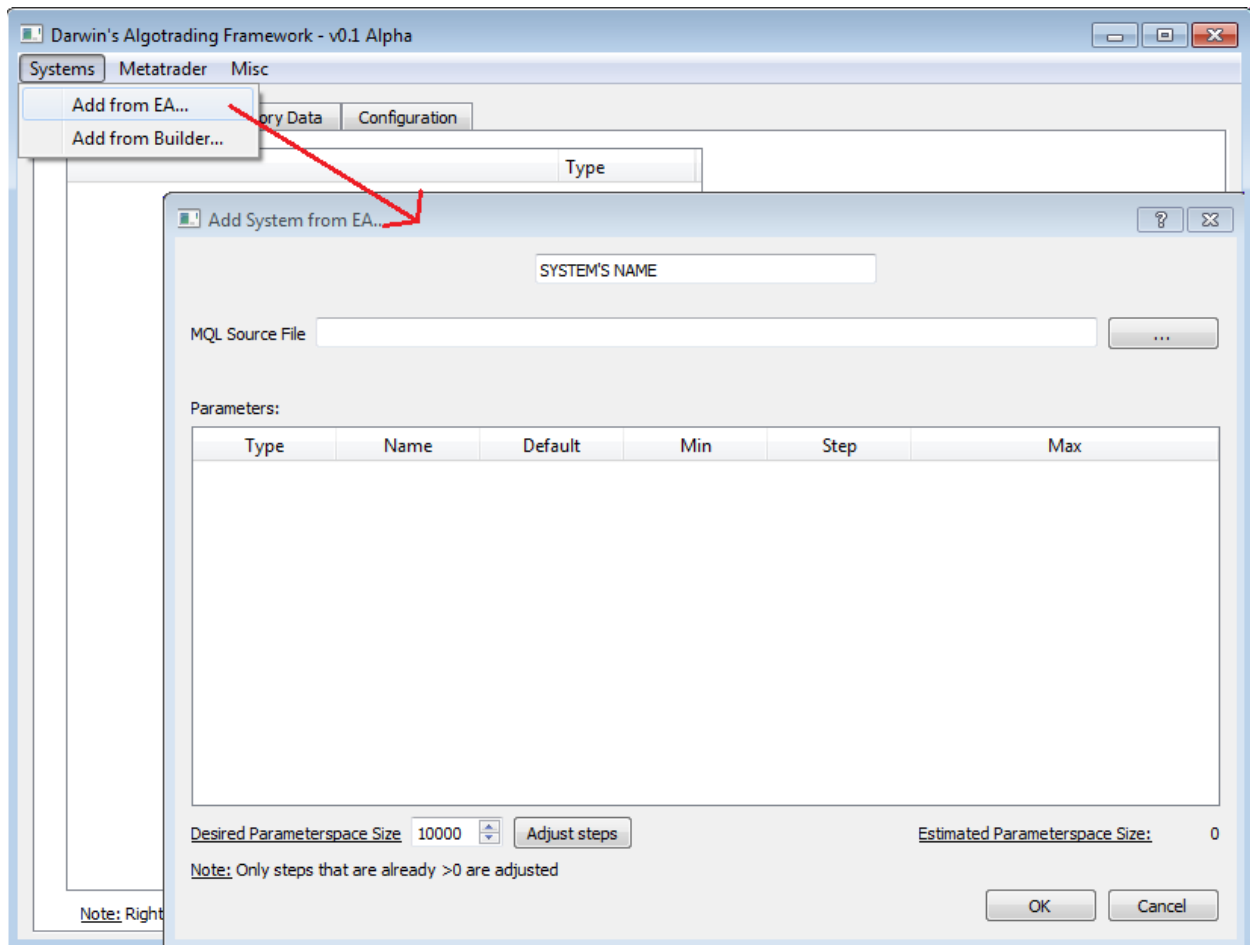
3. Open Metaeditor through DATFRA (do not open it manually. While you work with DATFRA, only start MT4 and Metaeditor through it, never manually!!)



4. Compile the EA that you want to analyse, just to check if it does so without errors. If it does, great!



5. Now you can close Metaeditor, go to DATFRA, and click 'Systems'-'>'Add from EA'



6. Decide on what parameter ranges you want to use. IMPORTANT: Every parameter that should be subject to optimisation needs to have a Step, only a min and max does not work!!

Add System from EA...

Test System

MQL Source File ...

Parameters:

Type	Name	Default	Min	Step	Max
double	pipValue	0.0001	0.0001		0.0001
double	lotsPerTrade	0.1	0.1		0.1
int	movingAveragePeriod	10	5	1	600
int	movingAverageType	0	0	1	3
int	whatPrice	0	0		0
int	reverseSignals	0	0	1	1

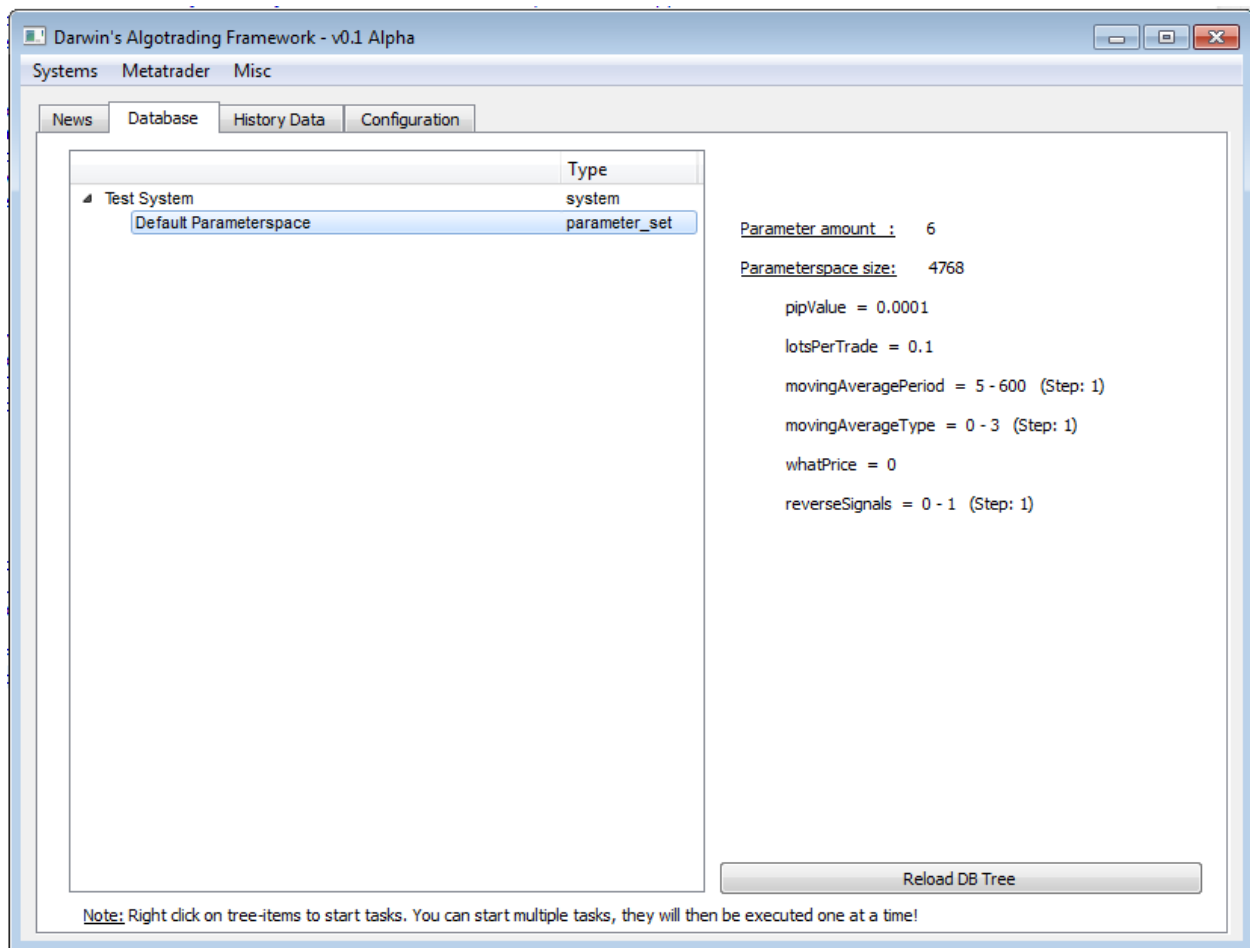
Desired Parameterspace Size Adjust steps

Estimated Parameterspace Size: 4768

Note: Only steps that are already >0 are adjusted

OK Cancel

7. You can now work with this EA in DATFRA



Note 0: If Metatrader4 wants an update (you see this when the User Account Control pops up during DATFRA-usage, asking you for write permission), start the MT4 that you use as basis for cluster building, let it update, close it and rebuild the cluster! Then this should work again :)

Note 1: When there is an update, DATFRA will tell you. Then, just download the new .exe, and replace the DATFRA.exe with the new one

Note 2: At the moment you can not stop Builder, Meta Analysis and PSA Data generation. If you want to stop these tasks, please go to 'Misc'->'Exit DATFRA, Stop All'

Note 3: DATFRA cleans its Expert directory. If it finds .ex4 or .log files without the corresponding source code (.mq1 or .mq4), it deletes it. Keep that in mind!

Note 4: Does not work? Kill DATFRA, and open 'debug_log_utils.txt' in DATFRA's main directory. You can, for example, see Metatrader Error reports in there.

Note 5: If there are errors during meta-analysis, like "mt4_instance::backtest ERROR, could not compile" or similar things, though the analysis itself is running and data-count increases, don't worry, everything is fine :)

Note 6: DO NOT DO MULTIPLE THINGS AT ONCE! If you are running a time-intensive task (Walk forward analysis, meta analysis, creation of parameterspace analysis data etc), leave it alone, and do not do other stuff with DATFRA at the same time. It might work, it might not.

1. Go to FOREXTESTER.COM and download their history data

Here you can download free history data for the most common currency pairs (Source: [Forexite. Ltd](http://Forexite.Ltd)):

Price: Bid

Time: GMT (no Daylight Saving Time)

Quality: one of the best from free sources

Symbol	Data Range	Size
AUDJPY	Jan 2001 - 30 Jun 2014	26.1 MB
AUDUSD	Jan 2001 - 30 Jun 2014	24.2 MB
CHFJPY	Jan 2001 - 30 Jun 2014	26.0 MB
EURCAD	Jan 2001 - 30 Jun 2014	29.0 MB
EURCHF	Jan 2001 - 30 Jun 2014	25.1 MB
EURGBP	Jan 2001 - 30 Jun 2014	22.4 MB
EURJPY	Jan 2001 - 30 Jun 2014	29.4 MB
EURUSD	Jan 2001 - 30 Jun 2014	26.9 MB
GBPCHE	Jan 2001 - 30 Jun 2014	31.0 MB
GBPJPY	Jan 2001 - 30 Jun 2014	31.7 MB
GBPUSD	Jan 2001 - 30 Jun 2014	27.7 MB

2. Extract them, open DATFRA and go to the 'History Data' Tab. Make sure to check the box 'Source: Forextester.com'

News

READ ME! VERY IMPORTANT!

Changelog

Trading System Database

History Data

Configuration

Symbol	Timeframe	Start	End
--------	-----------	-------	-----

Add from CSVs

☒ Source: forextester.com

☐ Dont generate M1 data

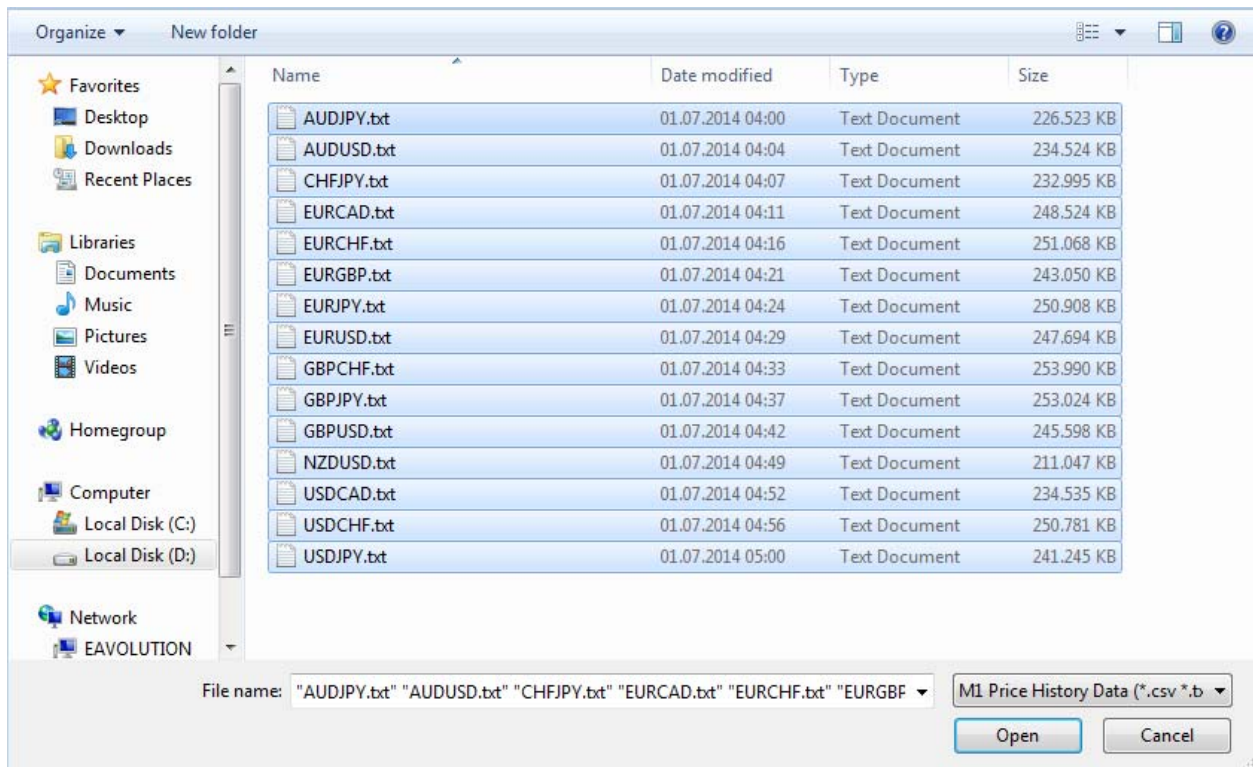
☐ Consistency check

Show me what data I have

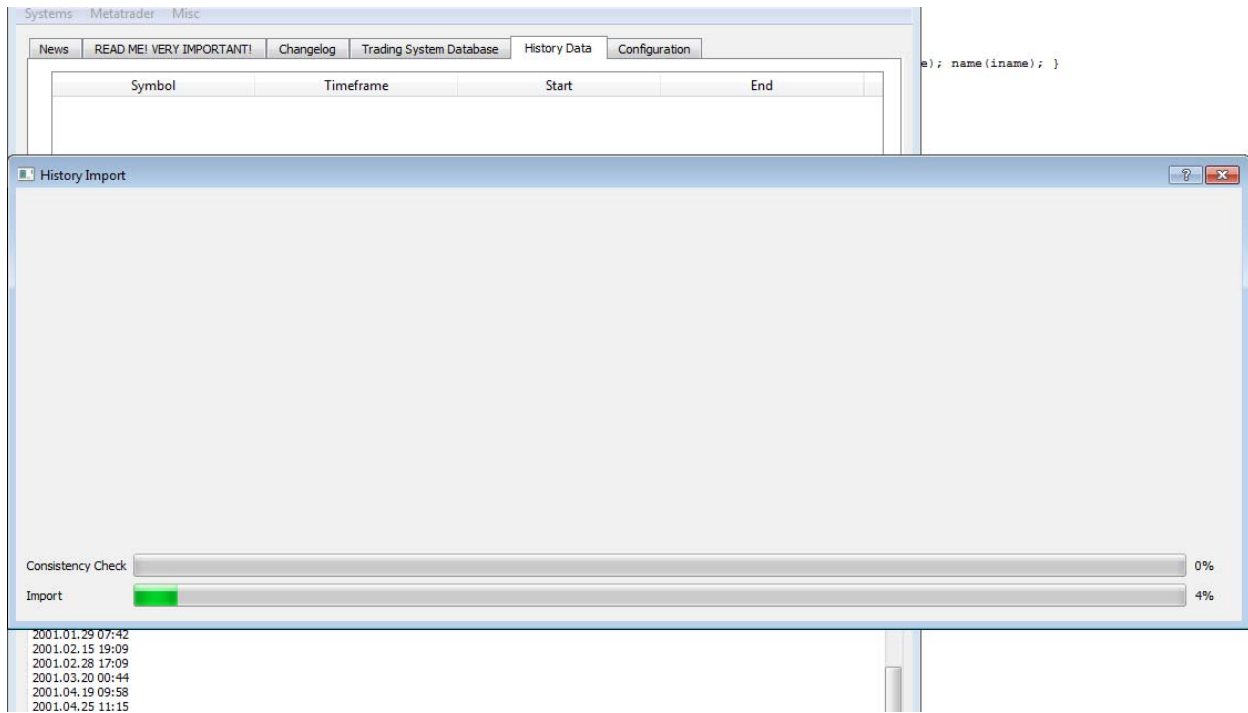
Truncate CSV

2013.01.01

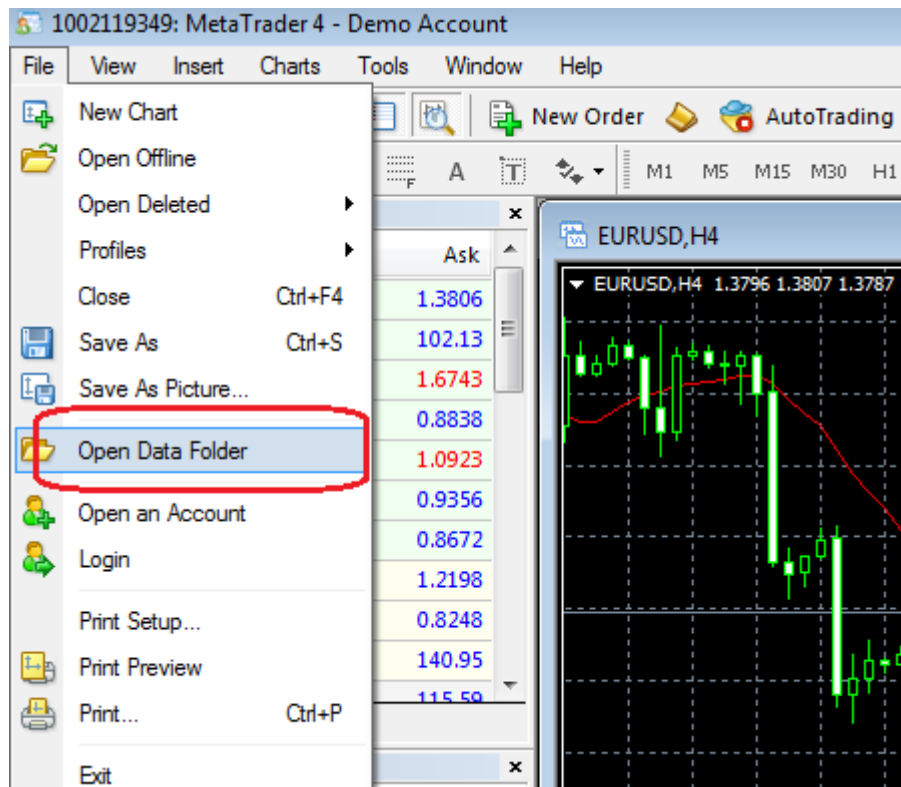
3. Click on 'Add from CSVs' and select them all.



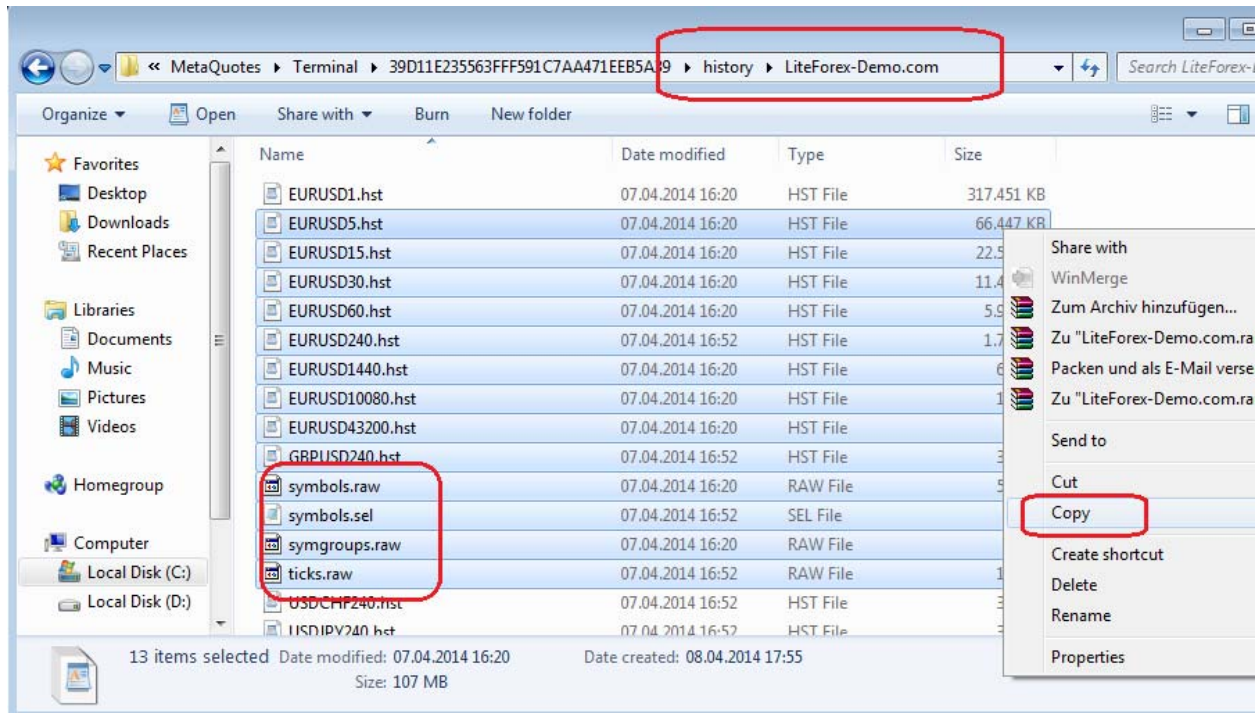
4. That's it. Now wait until DATFRA imported the data (there will be one of these windows for each dataset, one by one). When it's done, a message box will pop up, it will tell you to rebuild the MT4 cluster. Do so and you can use the new history data.



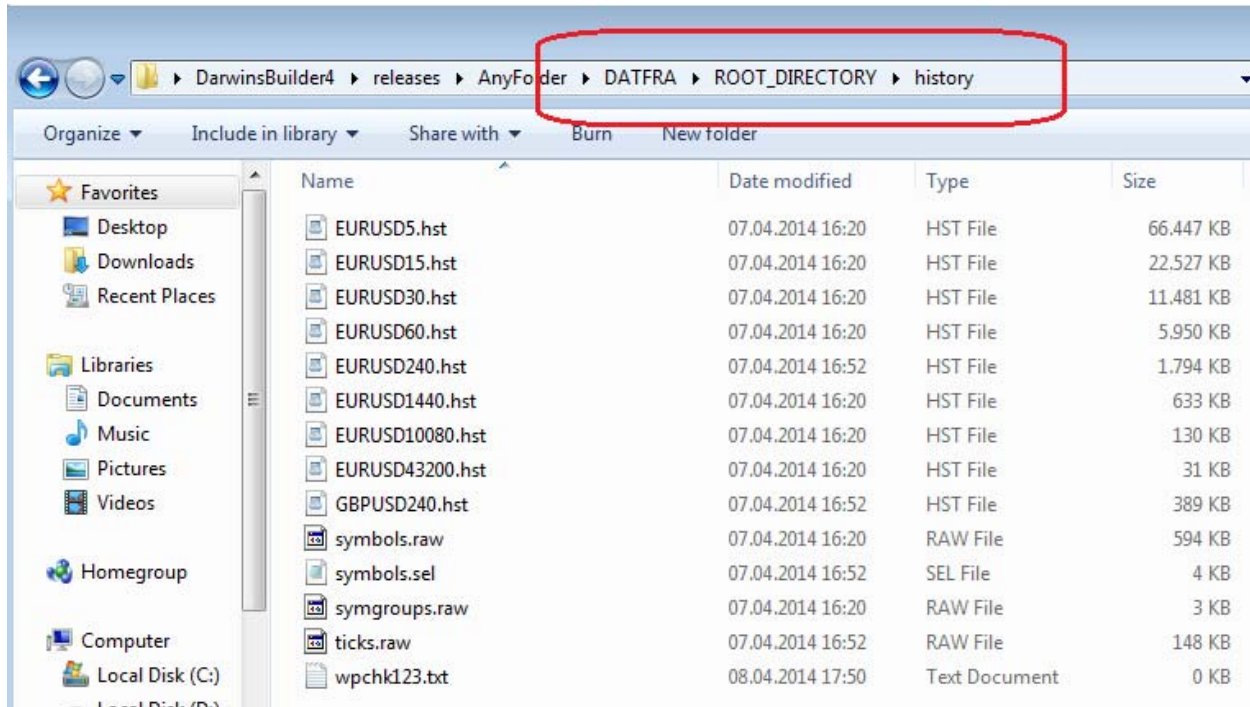
1. Open your regular Metatrader4, which already has the data you want to use, and click on 'Open Data Folder'



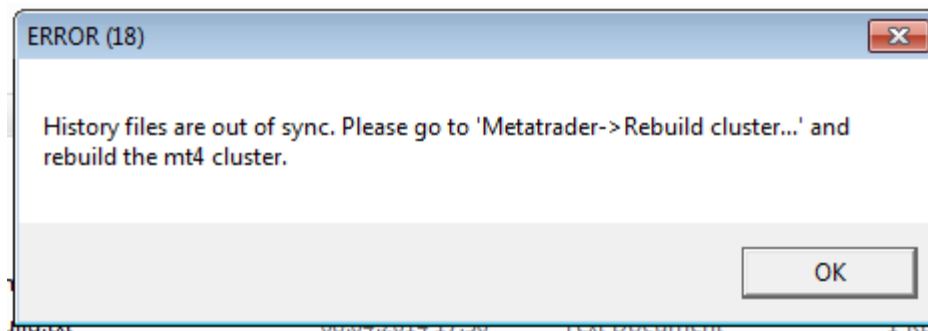
- Go to the history folder, then to the folder that holds the data (search if you don't know it), and then copy all .hst files you want to use in DATFRA, PLUS the .raw and .sel files!



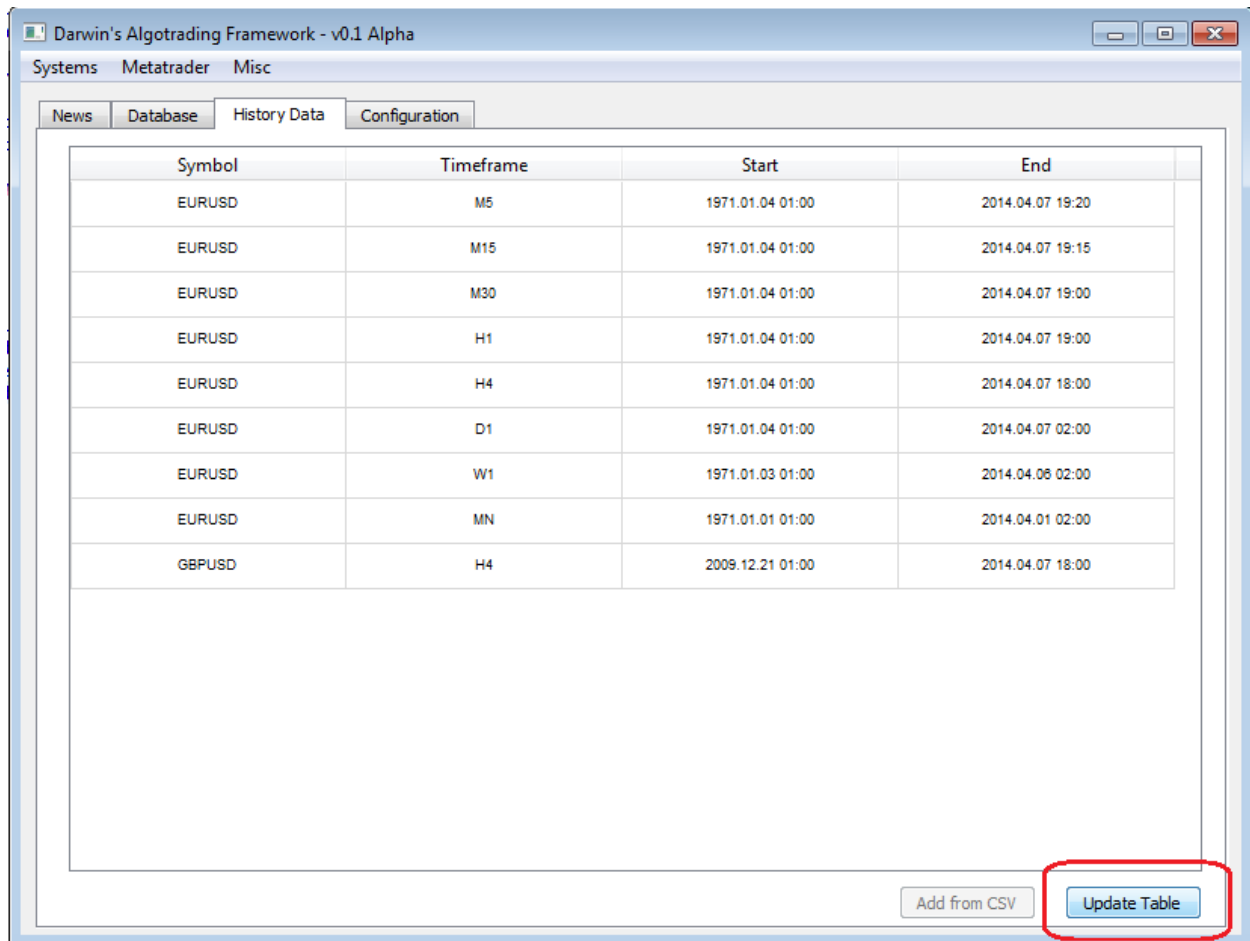
3. Go to the DATFRA directory, into ROOT_DIRECTORY/history and paste all files there. Then restart DATFRA!



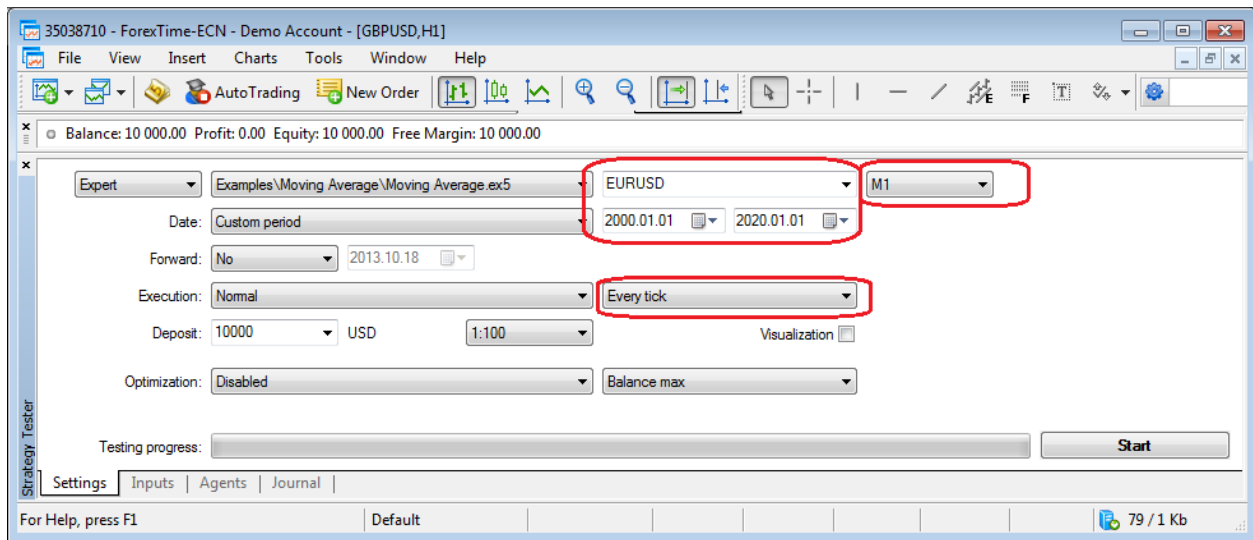
4. It will tell you that it did recognize the new files, and that you should rebuild the cluster. So, do it, like you already did once (Metatrader -> Rebuild cluster -> give it admin privileges -> press any key when you are told to -> done)



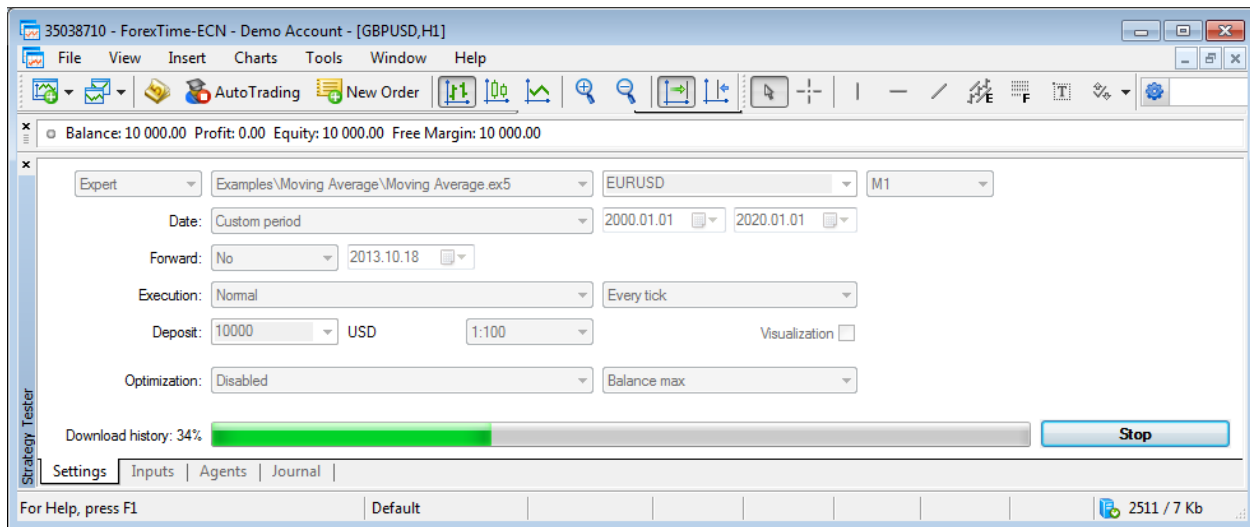
5. Go to the 'History Data' tab and click 'Update Table', to see if everything worked as intended.



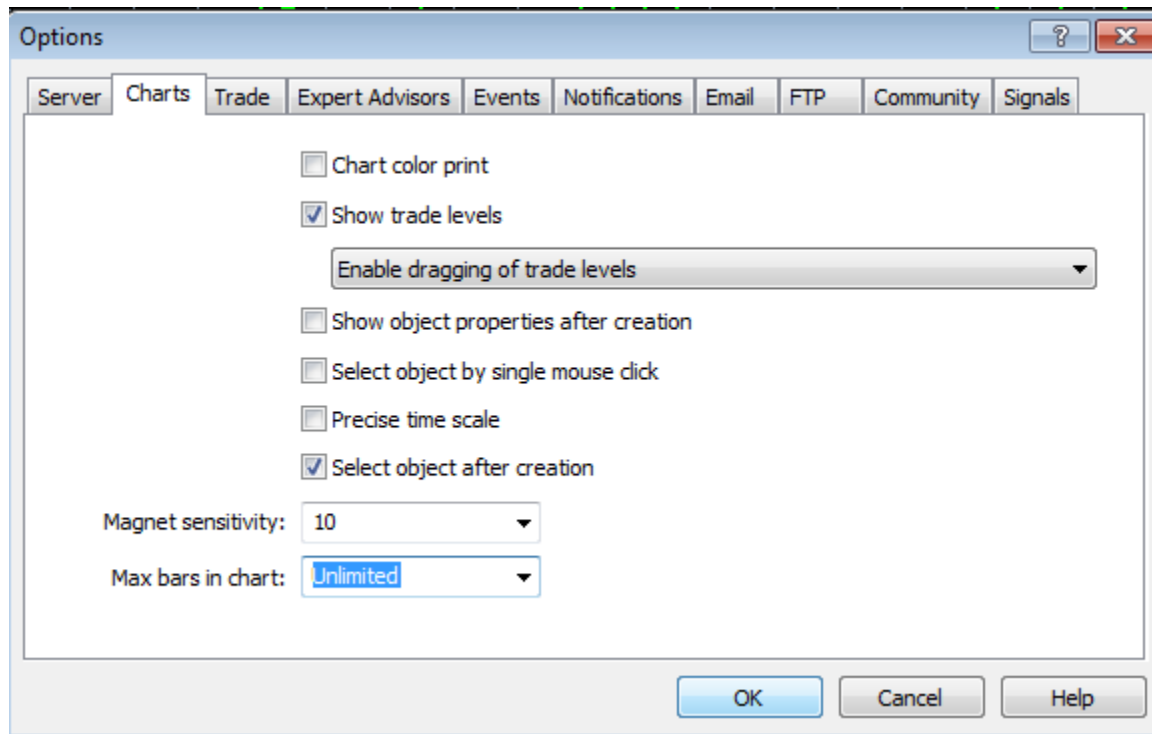
1. Open Metatrader5 and start a dummy backtest on M1(!), over the history data you want to have



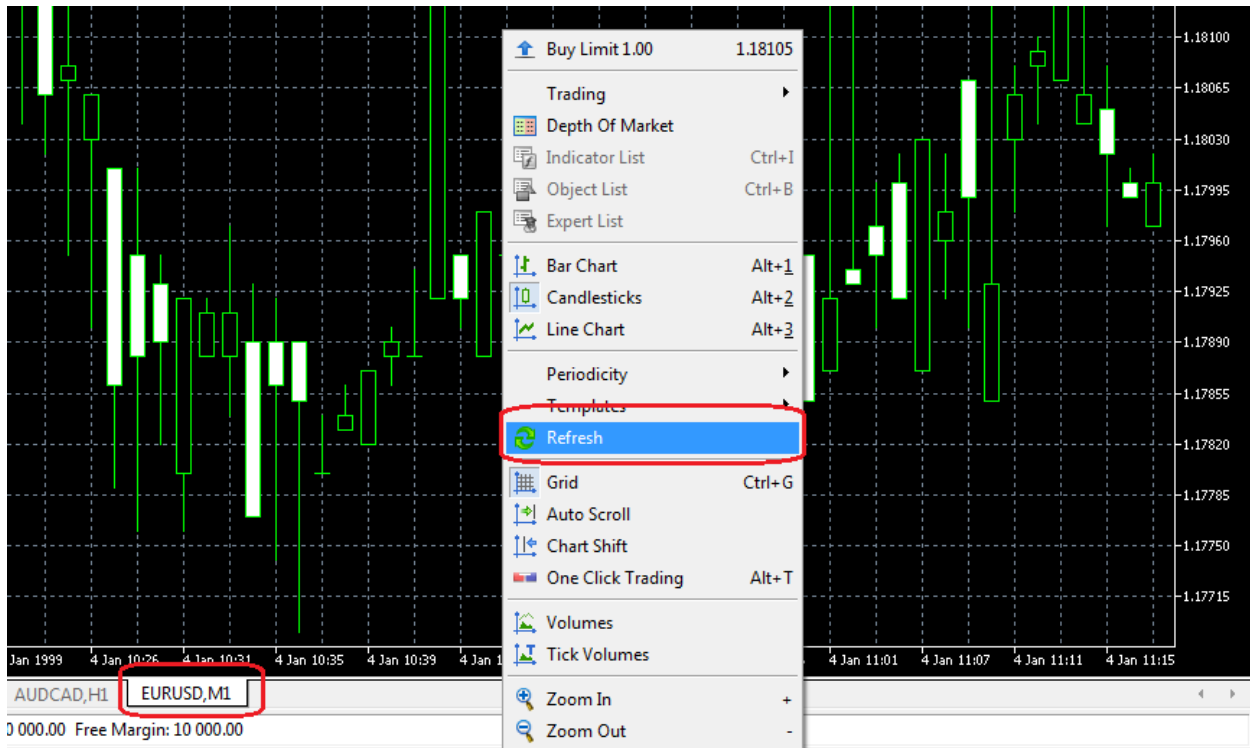
2. It will then automatically download the data. Note: MT5's data is a lot better than MT4's data!



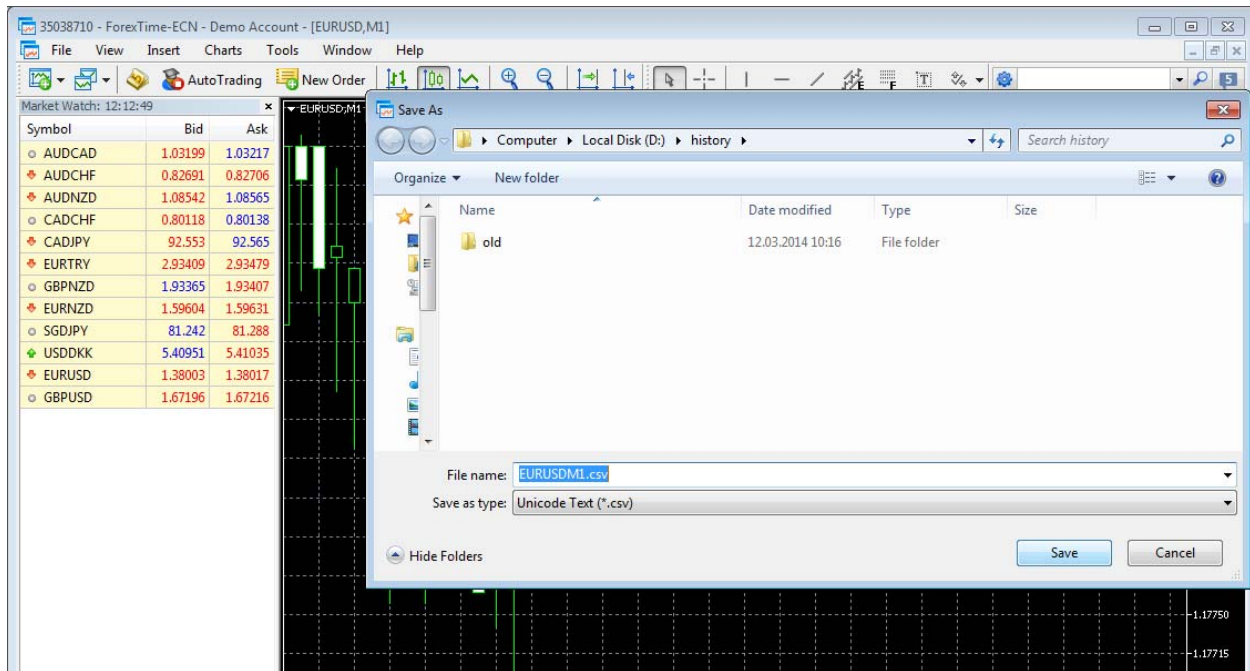
3. Afterwards, go to Options (Ctrl+O) and make sure you have 'Max Bars in Chart' set to 'Unlimited'



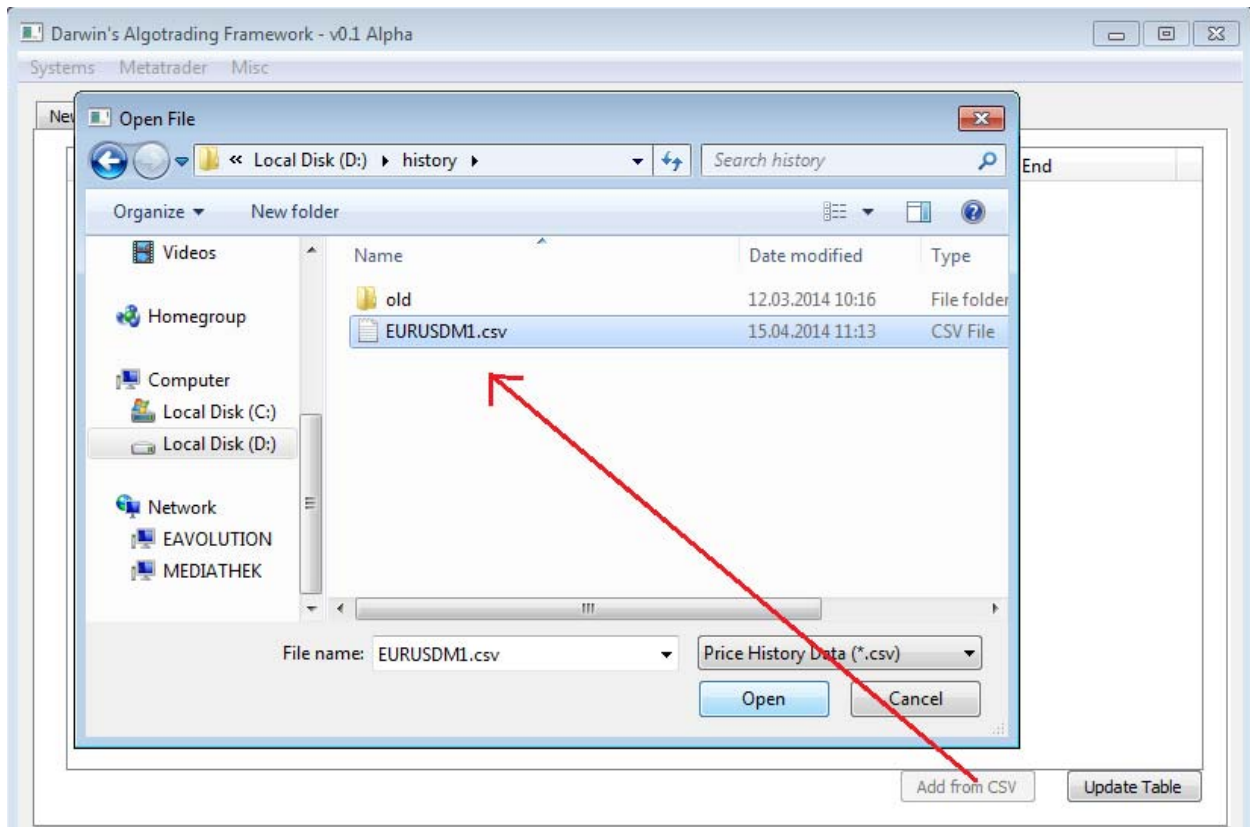
4. Then open the M1 Chart you just downloaded the data for. Make sure it goes back to the date you were targeting! If not, right click -> Refresh might help!



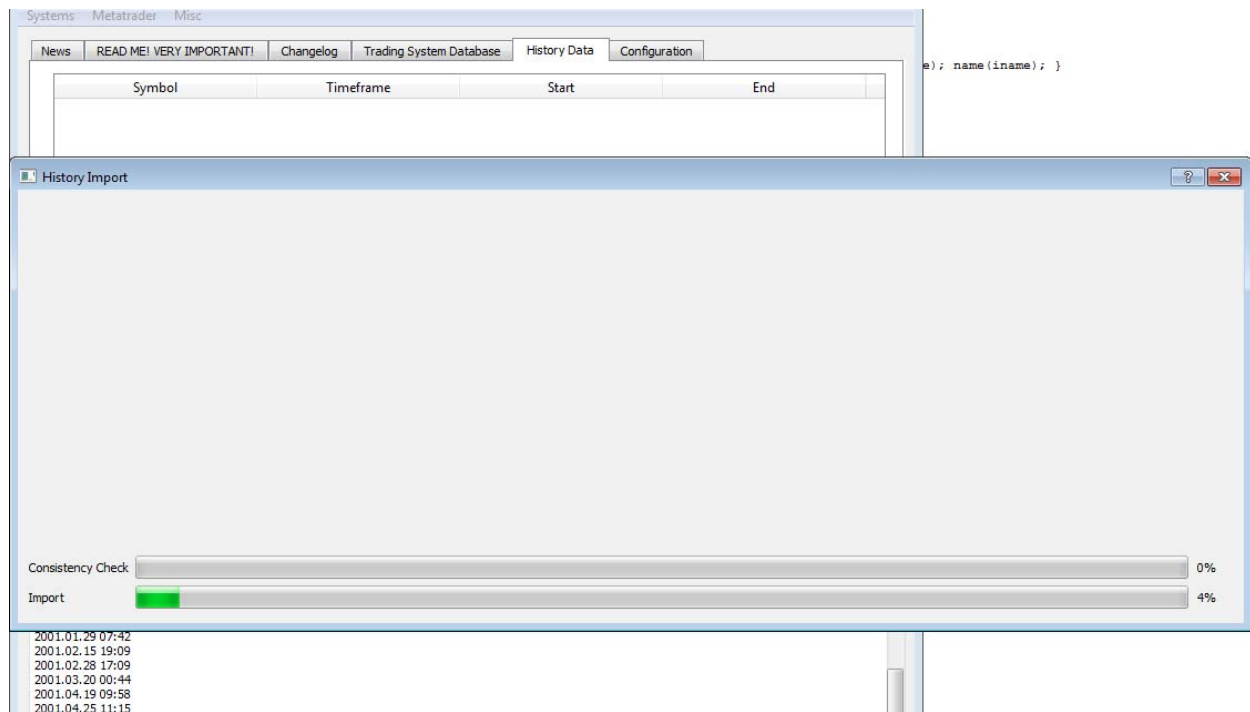
- Then press Ctrl+S, and it will ask you where to save the history data as csv file. The file will have ~400-700 Mb, check its size, to make sure everything worked



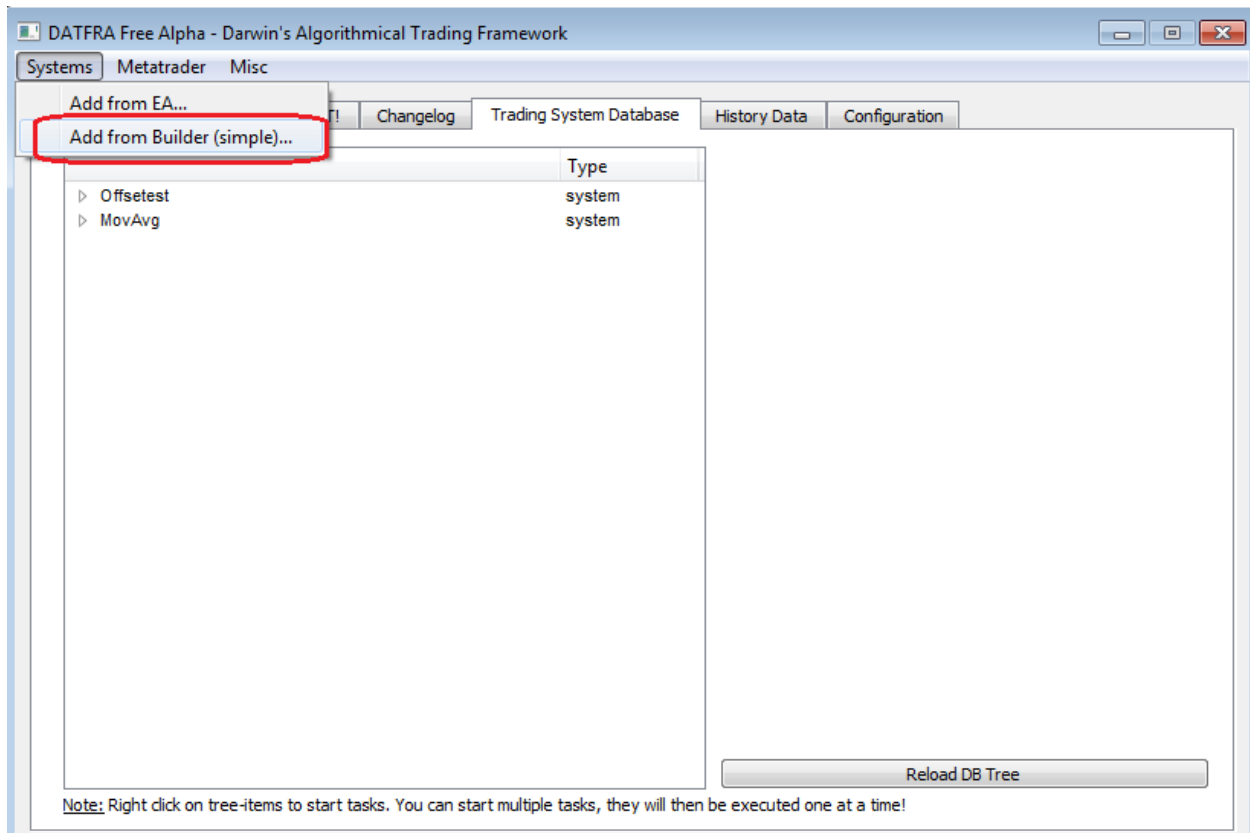
6. Close MT5, open DATFRA, go to the 'History Data' tab and load the just saved csv file.



7. That's it. Now wait until DATFRA imported the data (there will be one of these windows for each dataset, one by one). When it's done, a message box will pop up, it will tell you to rebuild the MT4 cluster. Do so and you can use the new history data.



1. Start the builder interface. IMPORTANT: DO NOT EXPECT DATFRA TO FIND GOOD SYSTEMS FREQUENTLY; THIS CAN TAKE DAYS OR WEEKS!



2. Select some indicators and the timeframes to use, both for entry and exit.

Builder Settings

System Build Settings System Test Settings

TEST BUILD

	Logical Functions	Timeframes	Indicators
ENTRY	<input checked="" type="checkbox"/> AND <input checked="" type="checkbox"/> OR <input type="checkbox"/> IFTHEN <input checked="" type="checkbox"/> NOT <input type="checkbox"/> GT / LT	M30 up to D1	<input type="checkbox"/> ADX <input type="checkbox"/> Ichimoku <input checked="" type="checkbox"/> MACD <input checked="" type="checkbox"/> Moving Averages <input checked="" type="checkbox"/> Price Action/High Lows <input type="checkbox"/> SAR <input type="checkbox"/> RSI <input type="checkbox"/> CCI <input type="checkbox"/> Stochastic Maximum Indicators per Entry: 3
EXIT	<input checked="" type="checkbox"/> AND <input checked="" type="checkbox"/> OR <input type="checkbox"/> IFTHEN <input checked="" type="checkbox"/> NOT <input type="checkbox"/> GT / LT	M15 up to H4	<input type="checkbox"/> ADX <input type="checkbox"/> Ichimoku <input type="checkbox"/> MACD <input checked="" type="checkbox"/> Moving Averages <input type="checkbox"/> Price Action/High Lows <input type="checkbox"/> SAR <input checked="" type="checkbox"/> RSI <input type="checkbox"/> CCI <input checked="" type="checkbox"/> Stochastic Maximum Indicators per Exit: 2

Randomly choose one method (Trailing or Fixed SL)

Start Strategy Building

The builder saves preview images of each generated system. To view them go to Misc -> Open previews of built strategies in the main window

3. Define the market on which the systems should be created. Leave enough out of sample data (otherwise your final WFA results will be biased!)

The screenshot shows the 'Builder Settings' window with two tabs: 'System Build Settings' and 'System Test Settings'. The 'Basic Settings' section is highlighted with a red box and contains the following fields:

- Main-Market: GBPUSD (dropdown)
- M30 (dropdown)
- Data: 01.01.2005 (calendar icon) to 01.01.2013 (calendar icon)
- Spread: 3 (spinners)
- Out of sample: 40 (spinners) % of data

Below these fields, a note states: "Out of sample data will not be used to create systems, it will be untouched at the time the WFA starts!".

The 'Additional WFA-tested Markets' section includes:

- Choose THIS one first (dropdown)
- Choose instrument first (dropdown)
- Add this market to list >> (button)

The 'Walk Forward Analysis Settings' section contains:

- Optimise on: 365 (spinners) - 365 (spinners) days of data,
- trade for max: 30 (spinners) days before reoptimisation.
- Preferred Characteristic: profit (dropdown)
- Guide optimisation towards: Genetic - Profit (dropdown)
- Choose from top: 50 (spinners) candidates.
- ☒ Don't pick one of them, forward-trade the AVERAGE parameterset of all these candidates.
- Minimum profit after WFA: 400 (spinners) pips / year

At the bottom, there is a 'Start Strategy Building' button and a note: "The builder saves preview images of each generated system. To view them go to Misc -> Open previews of built strategies' in the main window".

4. Here you can add other markets, and every found trading system will immediately be analysed on them. My recommendation: Leave it blank, let it build on one market (faster), and then decide manually which of the found systems is worth the effort to analyse it on further markets!

The screenshot shows the 'Builder Settings' window with two tabs: 'System Build Settings' and 'System Test Settings'. The 'Basic Settings' section includes fields for 'Main-Market' (GBPUSD), 'M30', 'Data' (01.01.2005 to 01.01.2013), 'Spread' (3), and 'Out of sample' (40 % of data). A note states: 'Out of sample data will not be used to create systems, it will be untouched at the time the WFA starts!'. The 'Additional WFA-tested Markets' section is highlighted with a red box and contains two dropdowns: 'Choose THIS one first' and 'Choose instrument first', followed by an 'Add this market to list >>' button. The 'Walk Forward Analysis Settings' section includes 'Optimise on' (365 days of data), 'trade for max' (30 days before reoptimisation), 'Preferred Characteristic' (profit), 'Guide optimisation towards' (Genetic - Profit), 'Choose from top' (50 candidates), a checked checkbox for 'Don't pick one of them, forward-trade the AVERAGE parameterset of all these candidates.', and 'Minimum profit after WFA' (400 pips / year). At the bottom, there is a 'Start Strategy Building' button and a note: 'The builder saves preview images of each generated system. To view them go to Misc -> Open previews of built strategies' in the main window'.

Builder Settings

System Build Settings System Test Settings

Basic Settings

Main-Market GBPUSD M30

Data 01.01.2005 01.01.2013

Spread 3

Out of sample 40 % of data

Out of sample data will not be used to create systems, it will be untouched at the time the WFA starts!

Additional WFA-tested Markets

Choose THIS one first

Choose instrument first

Add this market to list >>

Walk Forward Analysis Settings

Optimise on 365 - 365 days of data,

trade for max 30 days before reoptimisation.

Preferred Characteristic profit

Guide optimisation towards Genetic - Profit

Choose from top 50 candidates.

☒ Don't pick one of them, forward-trade the AVERAGE parameterset of all these candidates.

Minimum profit after WFA: 400 pips / year

Start Strategy Building

The builder saves preview images of each generated system. To view them go to Misc -> Open previews of built strategies' in the main window

5. Here you define how the optimisation is done. I recommend to leave the checkbox checked, as this works quite well against overfitting.

The screenshot shows the 'Builder Settings' window with two tabs: 'System Build Settings' and 'System Test Settings'. The 'Basic Settings' section includes fields for 'Main-Market' (GBPUSD), 'M30', 'Data' (01.01.2005 to 01.01.2013), 'Spread' (3), and 'Out of sample' (40 % of data). A note states: 'Out of sample data will not be used to create systems, it will be untouched at the time the WFA starts!'. The 'Additional WFA-tested Markets' section has dropdowns for 'Choose THIS one first' and 'Choose instrument first', and a button 'Add this market to list >>'. The 'Walk Forward Analysis Settings' section includes 'Optimise on' (365 days of data), 'trade for max' (30 days before reoptimisation), 'Preferred Characteristic' (profit), 'Guide optimisation towards' (Genetic - Profit), 'Choose from top' (50 candidates), a checked checkbox for 'Don't pick one of them, forward-trade the AVERAGE parameterset of all these candidates.', and 'Minimum profit after WFA' (400 pips / year). A 'Start Strategy Building' button is at the bottom, with a note: 'The builder saves preview images of each generated system. To view them go to Misc -> Open previews of built strategies' in the main window'.

Builder Settings

System Build Settings System Test Settings

Basic Settings

Main-Market GBPUSD M30

Data 01.01.2005 01.01.2013

Spread 3

Out of sample 40 % of data

Out of sample data will not be used to create systems, it will be untouched at the time the WFA starts!

Additional WFA-tested Markets

Choose THIS one first

Choose instrument first

Add this market to list >>

Walk Forward Analysis Settings

Optimise on 365 - 365 days of data,

trade for max 30 days before reoptimisation.

Preferred Characteristic profit

Guide optimisation towards Genetic - Profit

Choose from top 50 candidates.

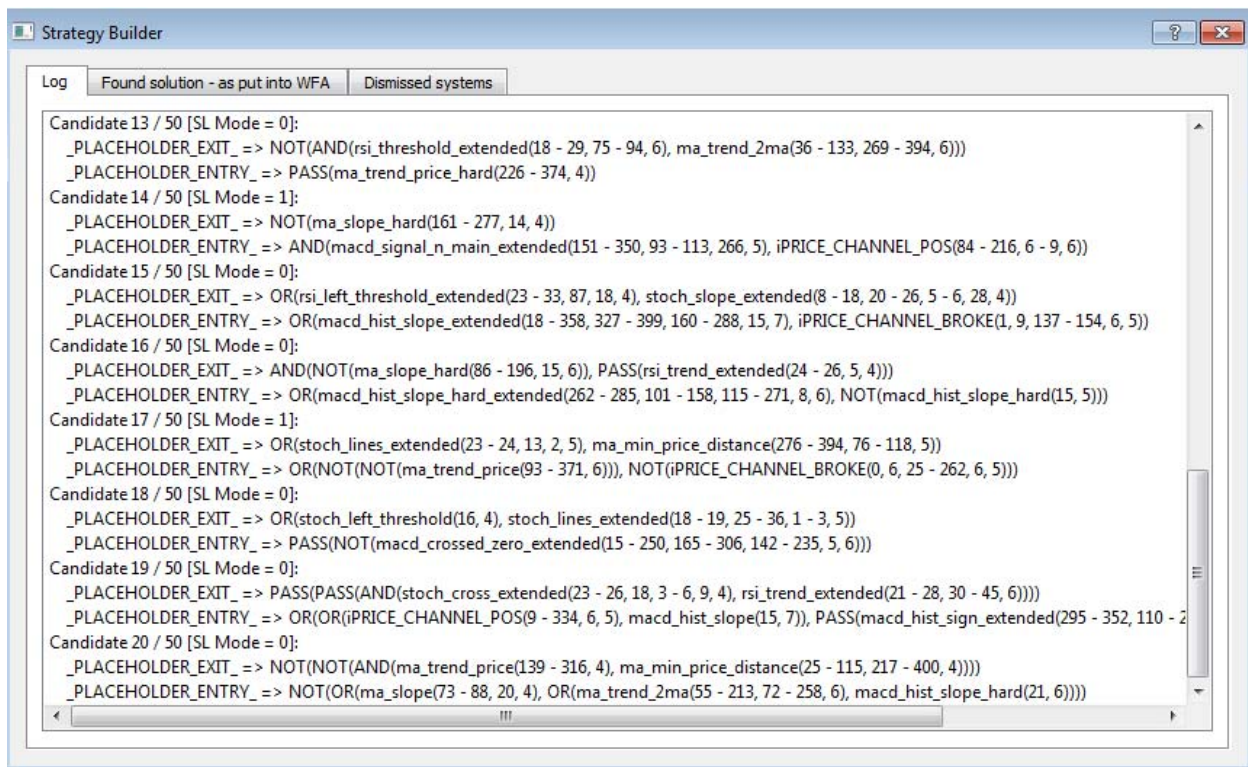
☒ Don't pick one of them, forward-trade the AVERAGE parameterset of all these candidates.

Minimum profit after WFA: 400 pips / year

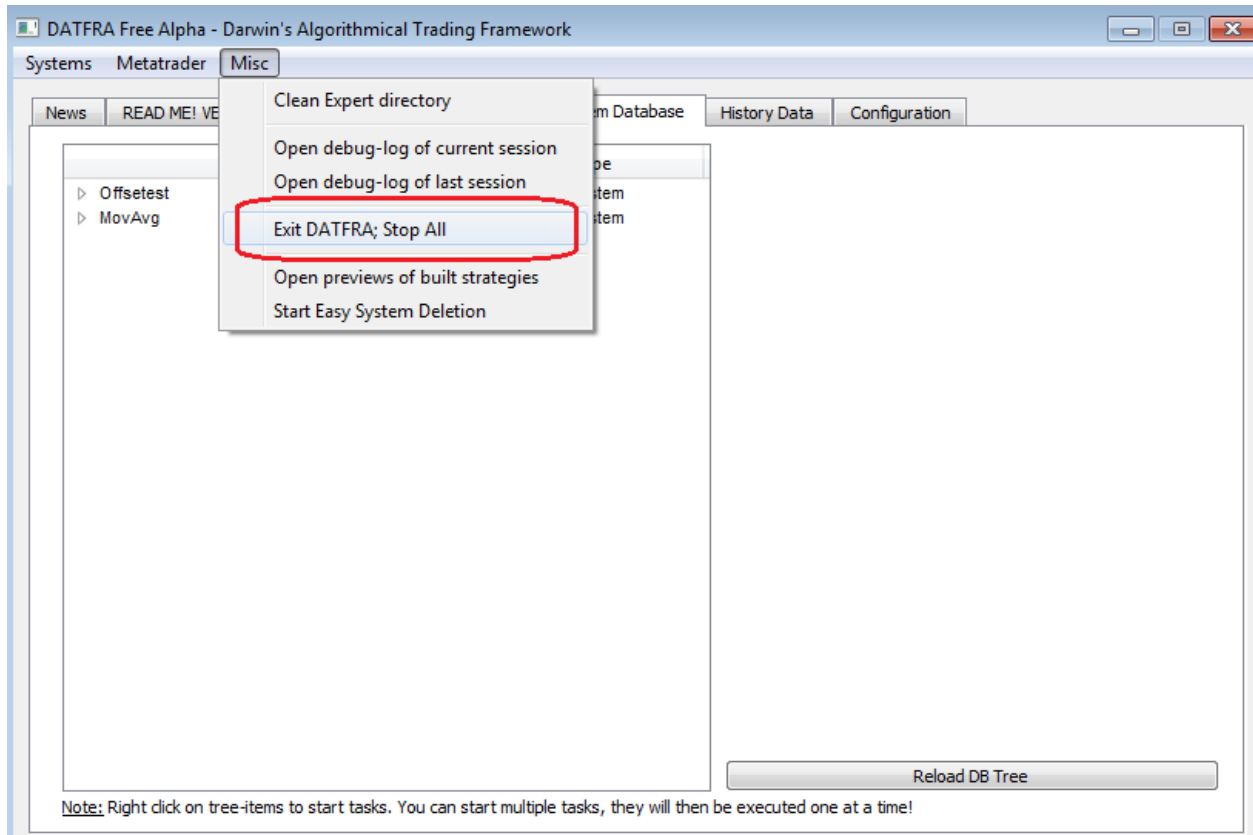
Start Strategy Building

The builder saves preview images of each generated system. To view them go to Misc -> Open previews of built strategies' in the main window

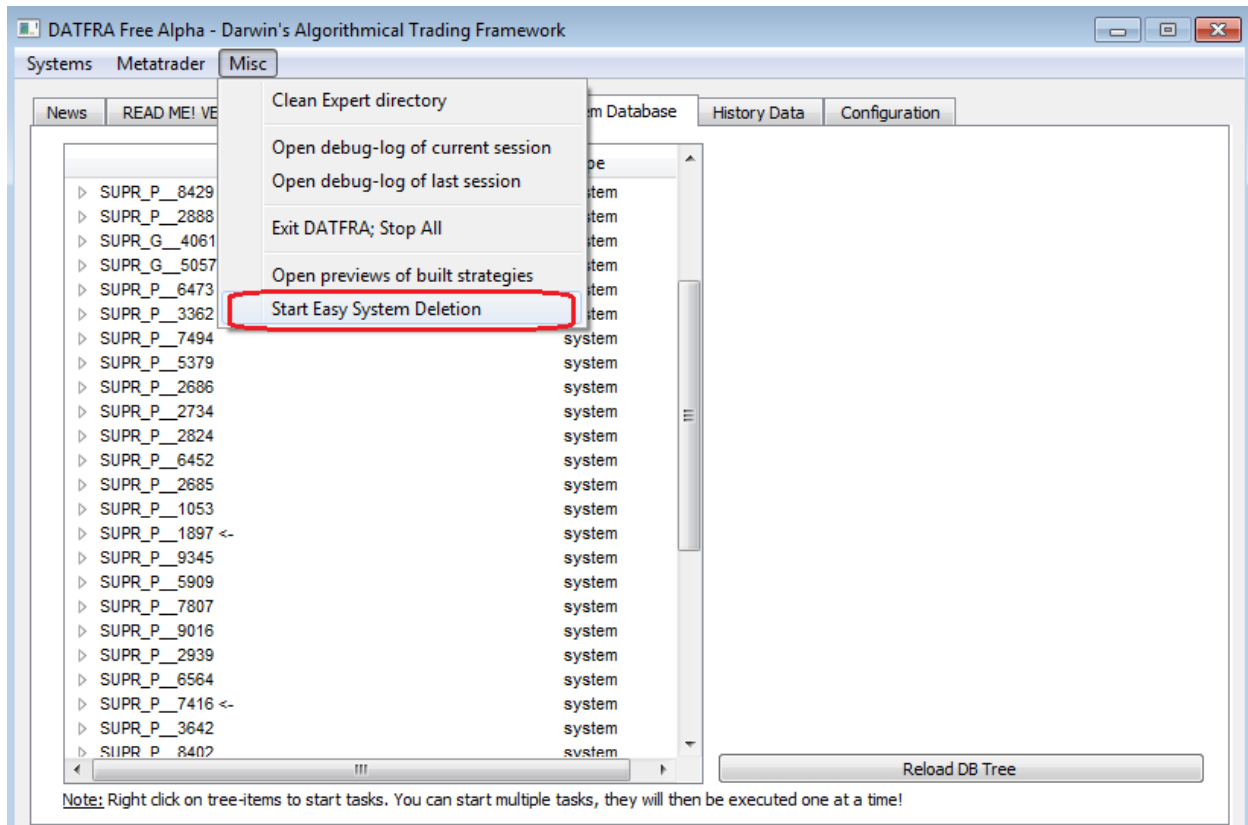
6. Now let it just build for as long as you want. When you have enough...



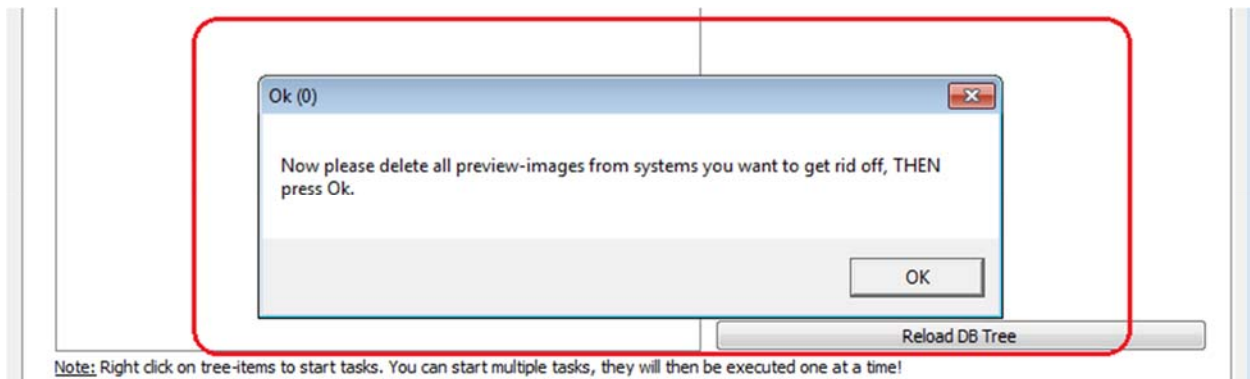
7. ... end DATFRA with this button! That will effectively terminate the building process. Just closing the windows will most likely let DATFRA keep running in the background (thats a bug, and it wont get fixed until the more important stuff is done!)



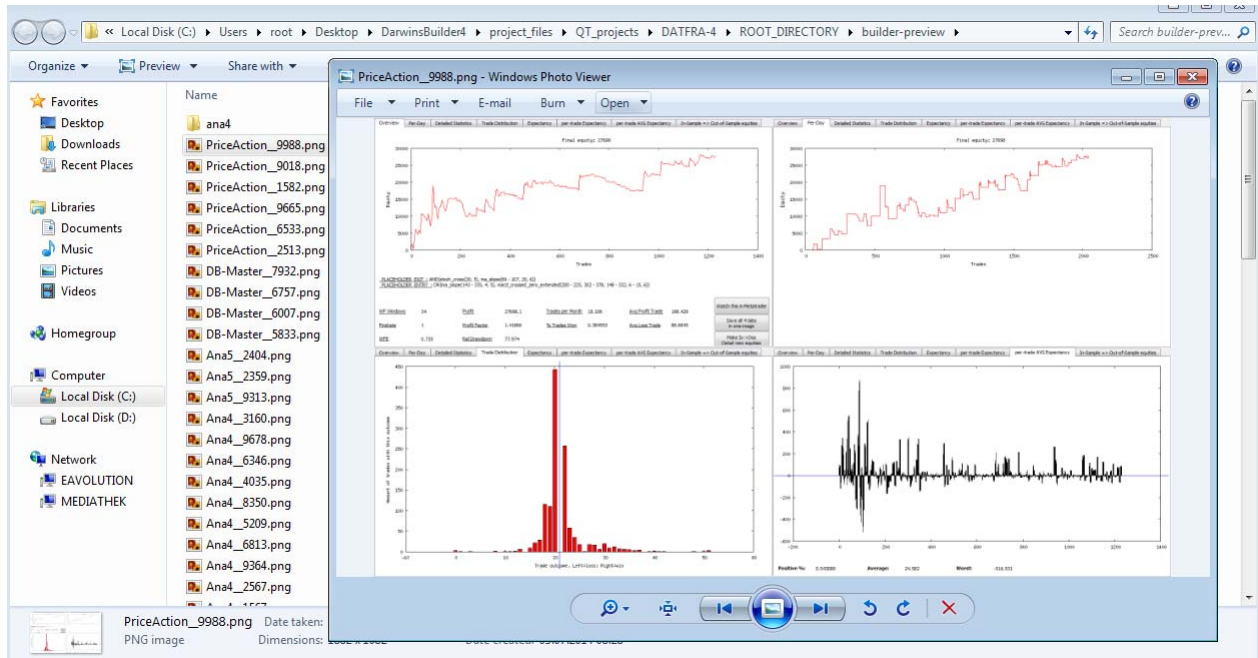
1. After a building session (usually at least 12-24 hours!), you will have some systems in your database, and you want an easy way to get rid of the losers, so click on this button



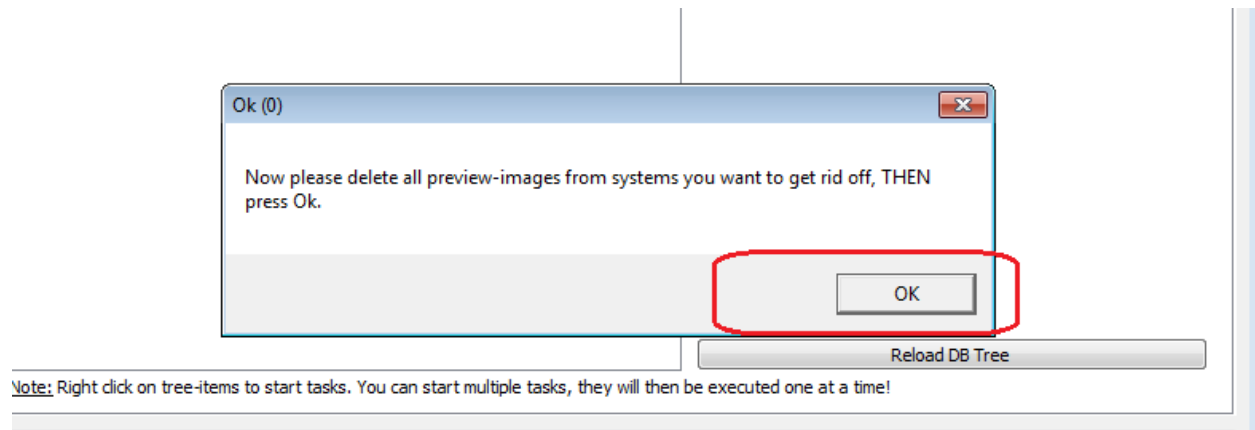
2. Now a directory will open and this message box. DO NOT CLICK ON OK! Go to the opened directory...



3. ..and open the first image in windows photo viewer. You can then cycle through all images with the arrow keys, and delete an image with 'Del' key. Every system's preview image you delete here, will afterwards be deleted from your DATFRA database!



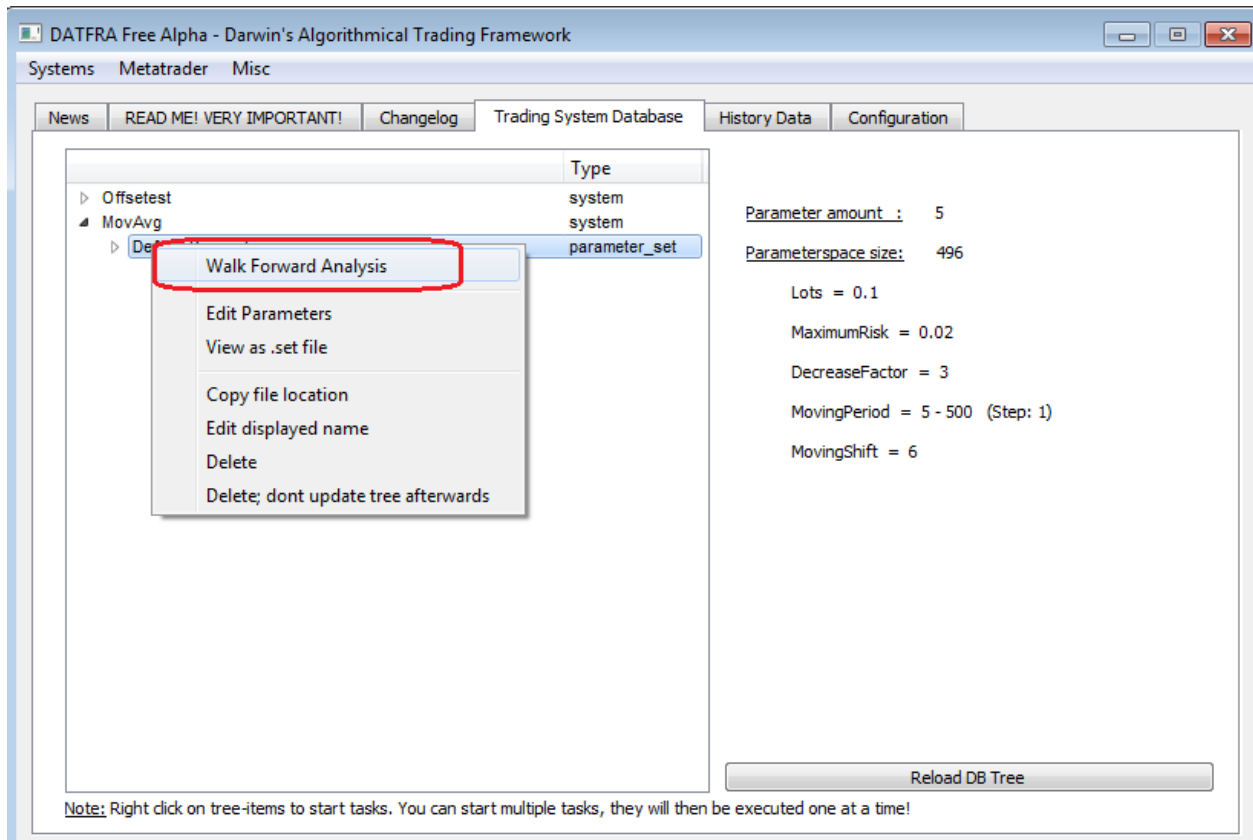
4. When you are done deleting images, click on the 'OK' of the previously opened message box. Now you will see that DATFRA will delete the corresponding systems.



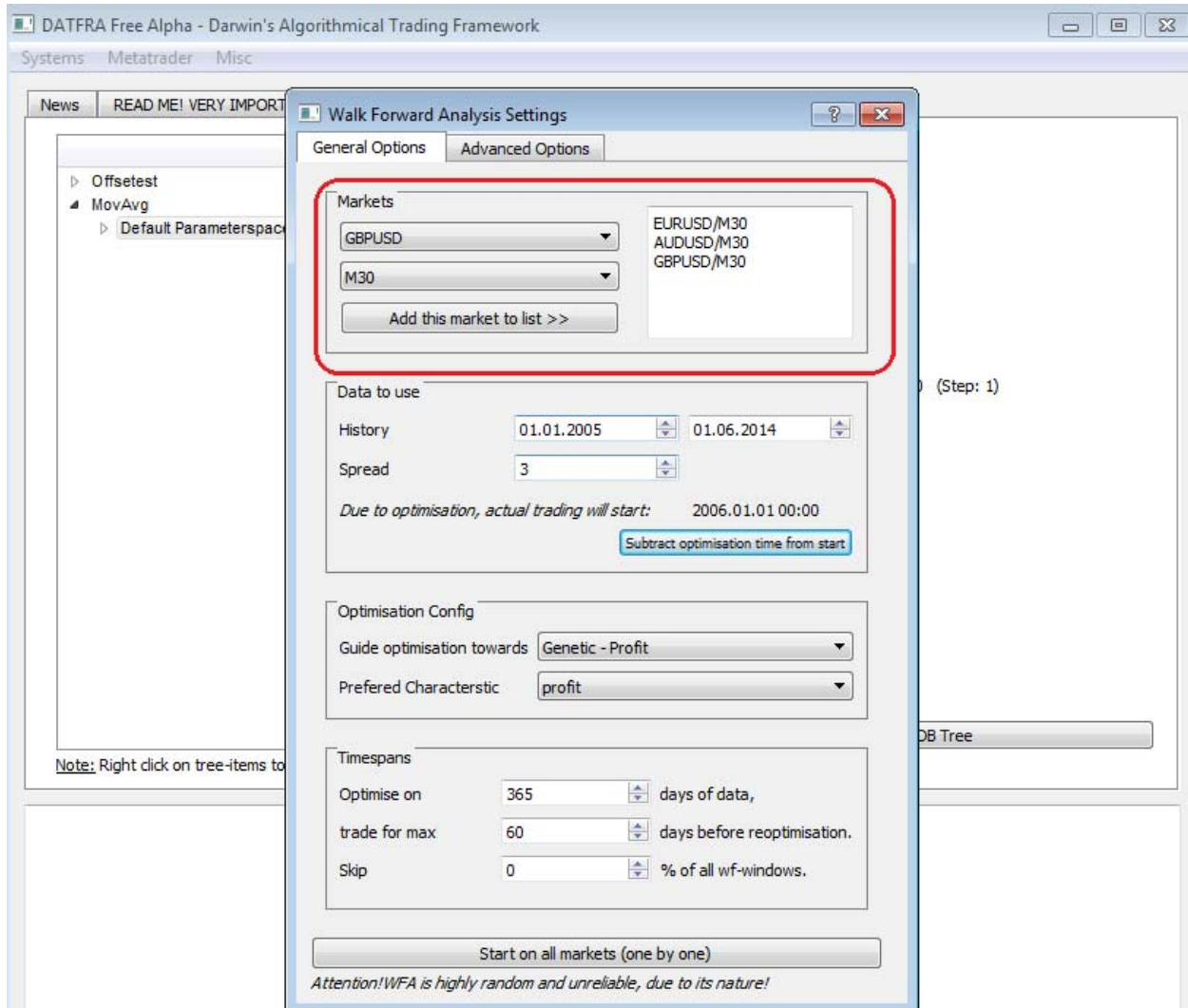
5. Now its your turn - do your job as a trader! Tip: I usually start analysing the found systems on other markets - and the most of them will fail this test. Only the very very best will be profitable on multiple markets - but these are the ones we want. However, that does not mean others could not succeed. An 'Endless WFA' is another possibility to test a found system in a more in-depth way!

=)

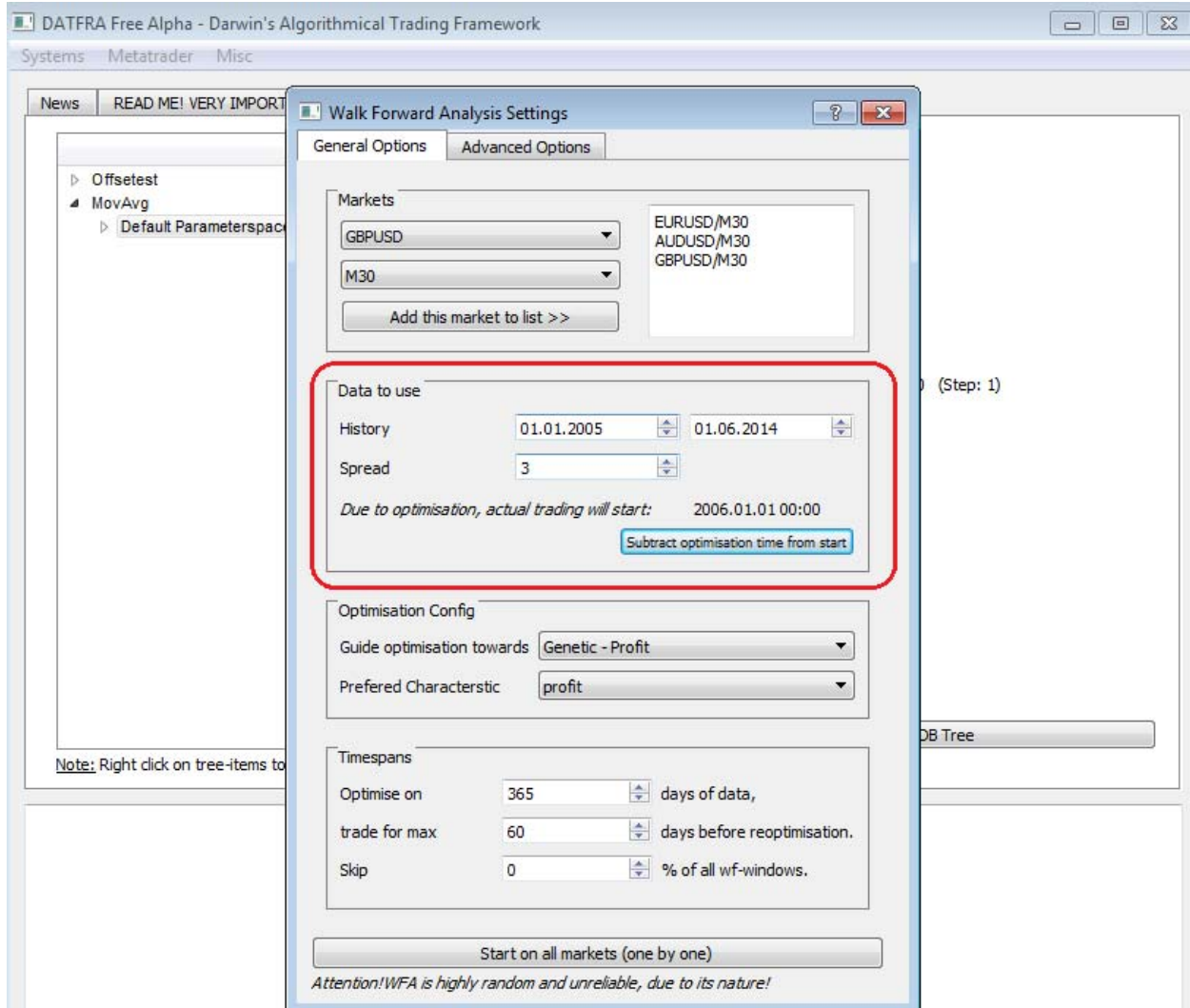
1. Start a Walk Forward Analysis on any parameterspace



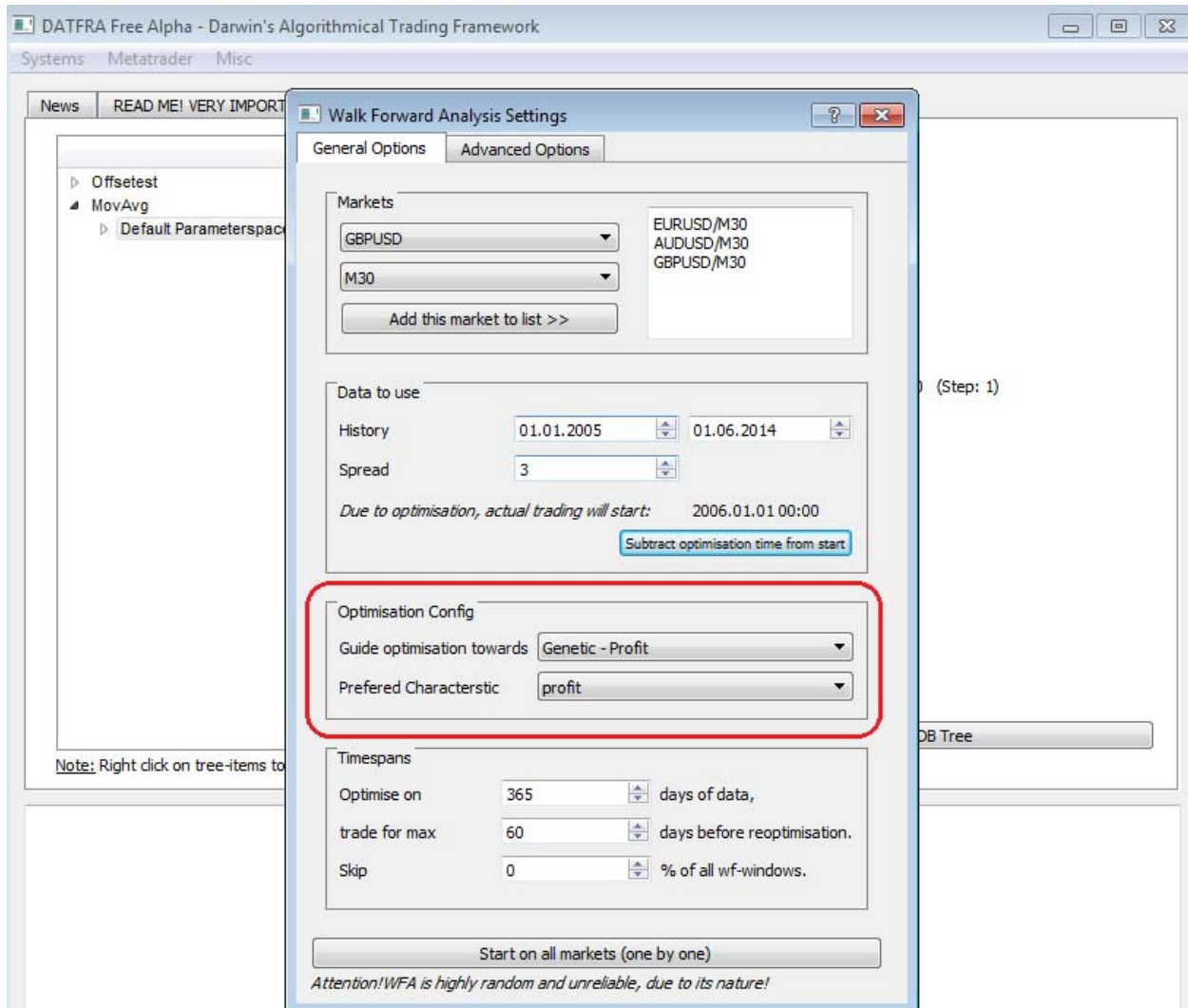
2. First, add every market you want to analyse to the list. A WFA will be started for each of them separately, one by one!



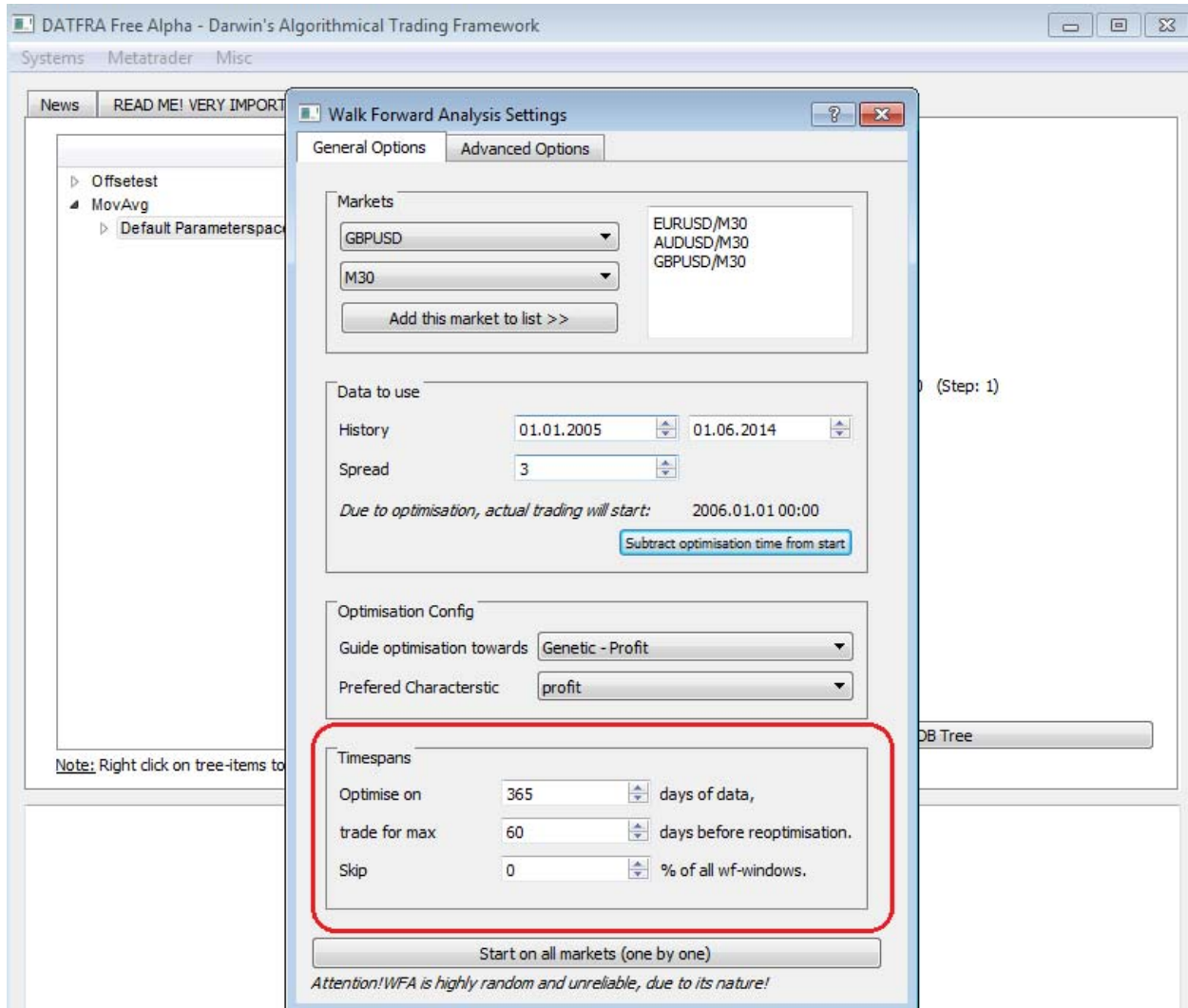
3. Start and End Date. Note: This defines the total data to use. So if your optimisation timespan is 365 days, for example, the first year of the data will not be used for trading, but only for optimisation. If you want to start actual trading on this date, press 'Subtract optimisation time from start' button.



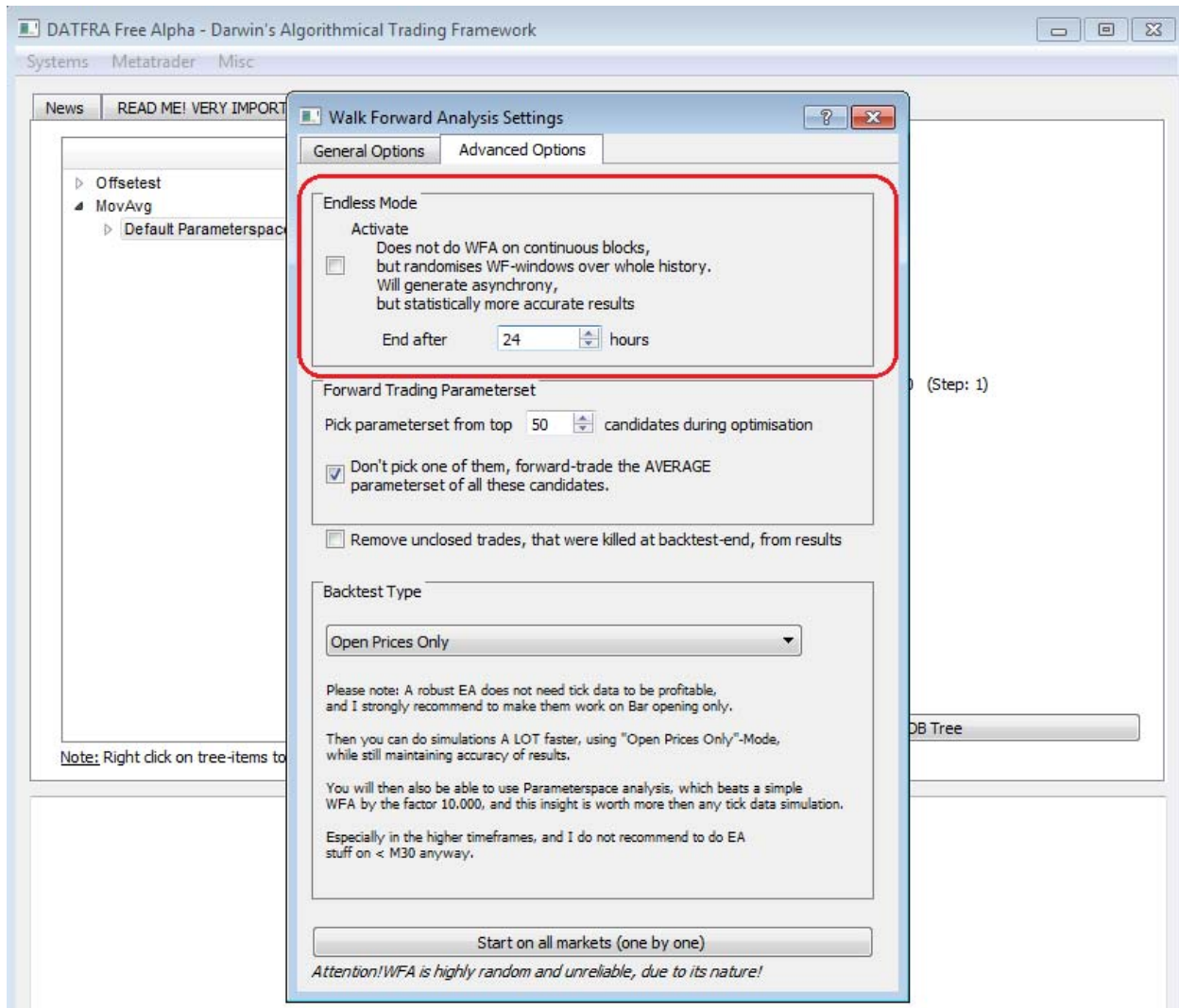
4. How Metatrader4 will do the optimisation. These settings are recommended, they are fast and give good results. Whole Parameterspace is not recommended, it takes too long and you will end up with lot of overfitted solutions.



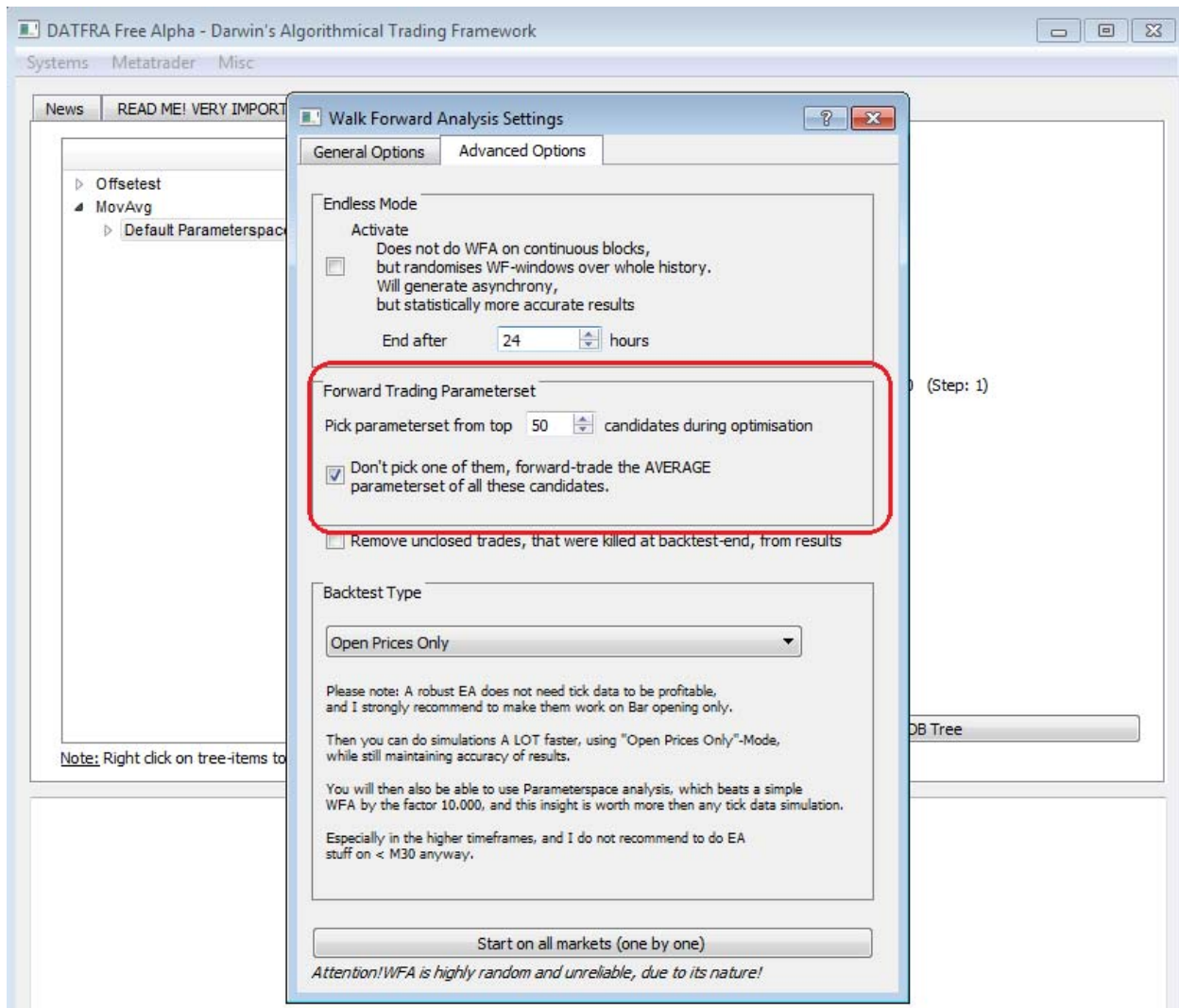
- Define how often the EA should be re-optimised, and on how much past-data it should be re-optimised



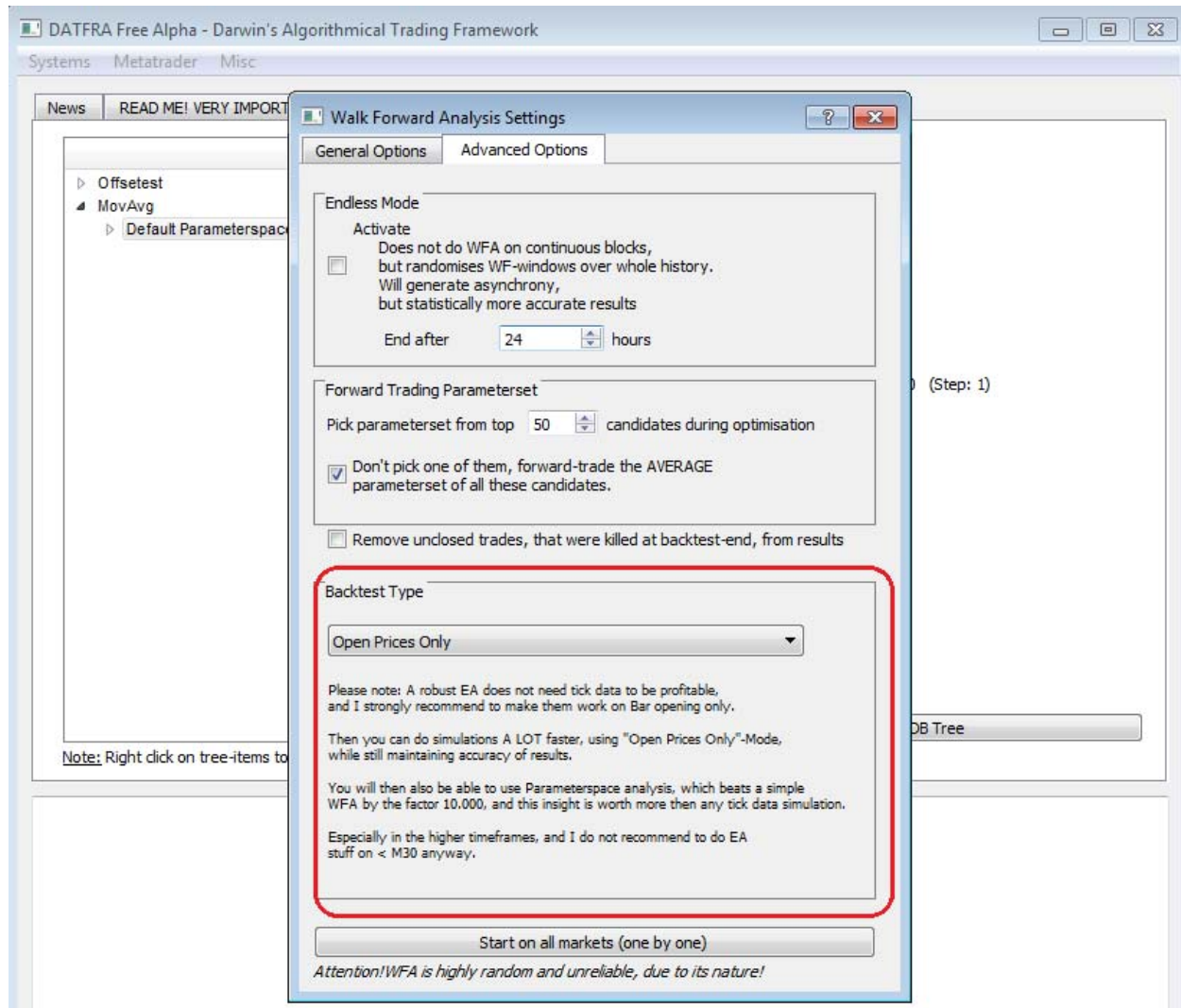
6. This mode gets the most out of your available history data, it will randomly take some point within the history as 'start' point for next WFA-window each time. This will give you most realistic characteristics of a system, but will also take a lot of time. Its a very nice option if you want an in-depth look on a system!



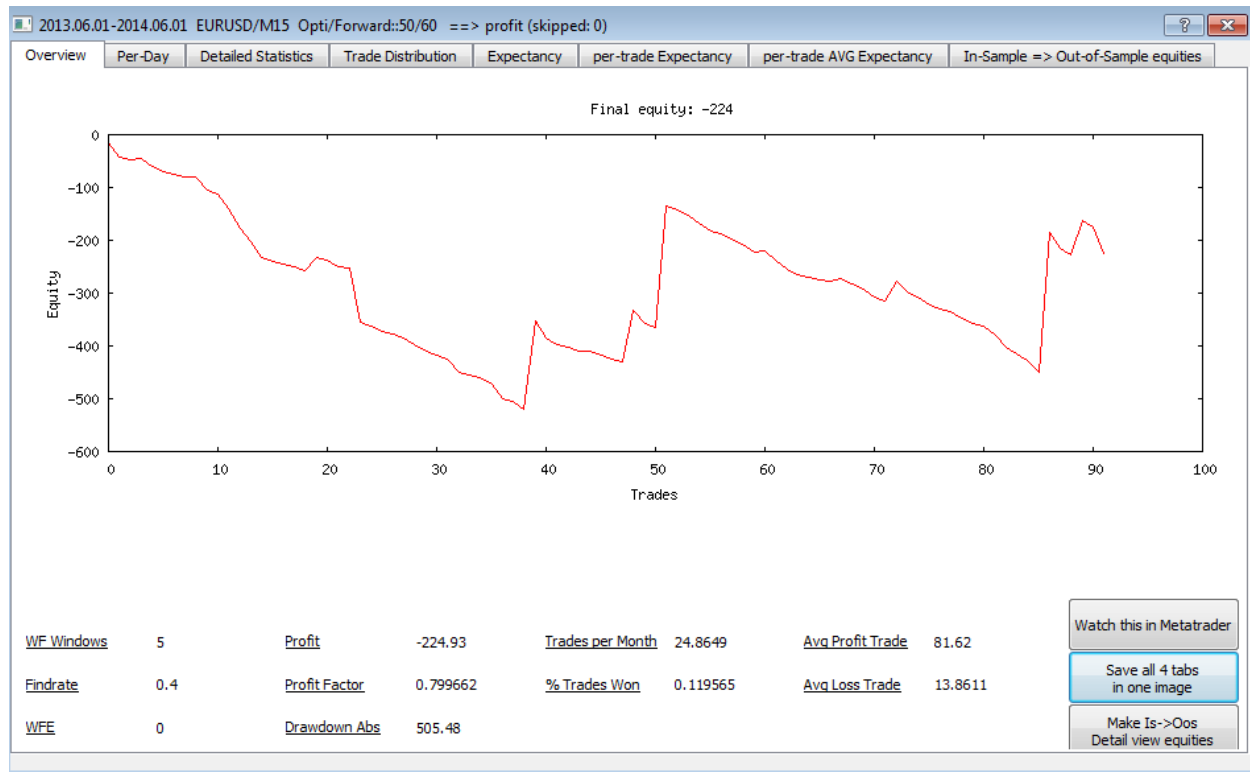
7. Recommended option! This will not only forward-trade the best parameterset of an optimisation, but take the best X parameters, then average them, and then trade this averaged parameterset forward.



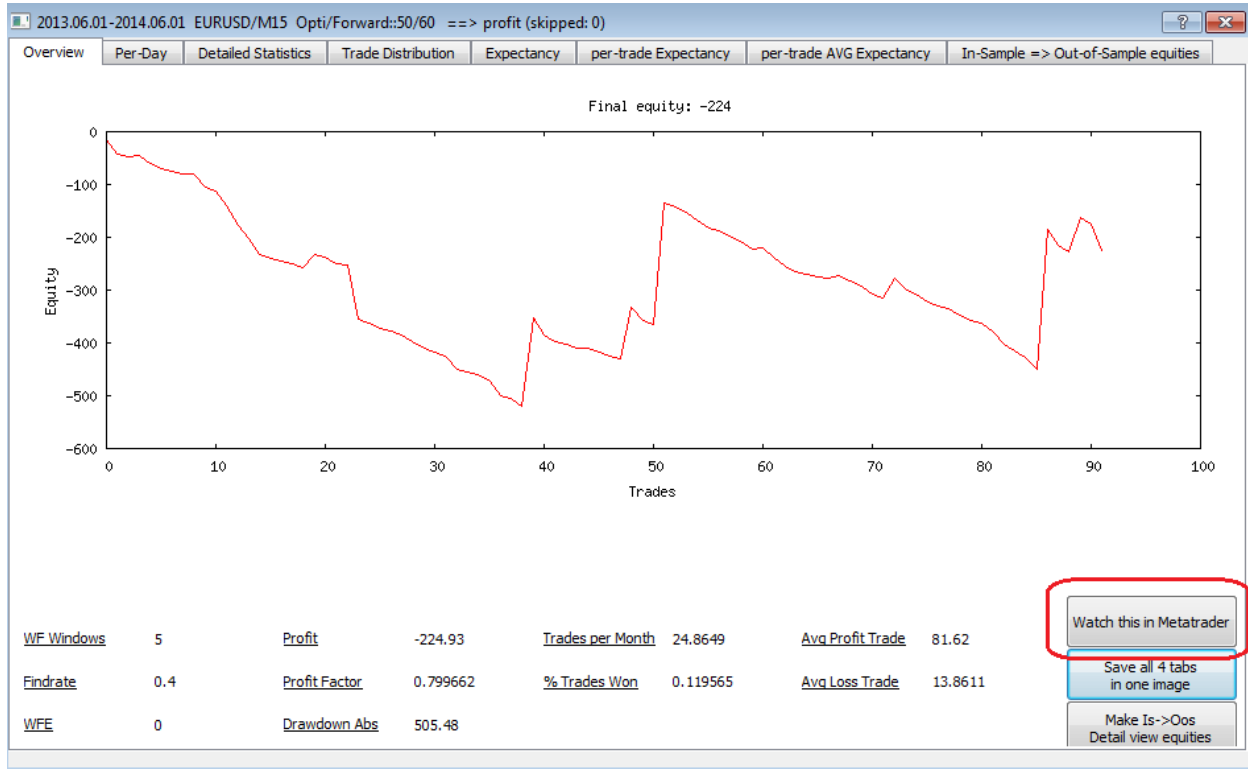
8. You know this option from Metatrader4. I strongly recommend to develop EAs to work on 'Bar opening only' if you want to do algorithmic trading, otherwise a single backtest takes way too long and you will, from simple time related reasons, not be able to use more sophisticated algorithms and analysis methods. Heck, even WFA will take ages with 'Every tick' backtests.



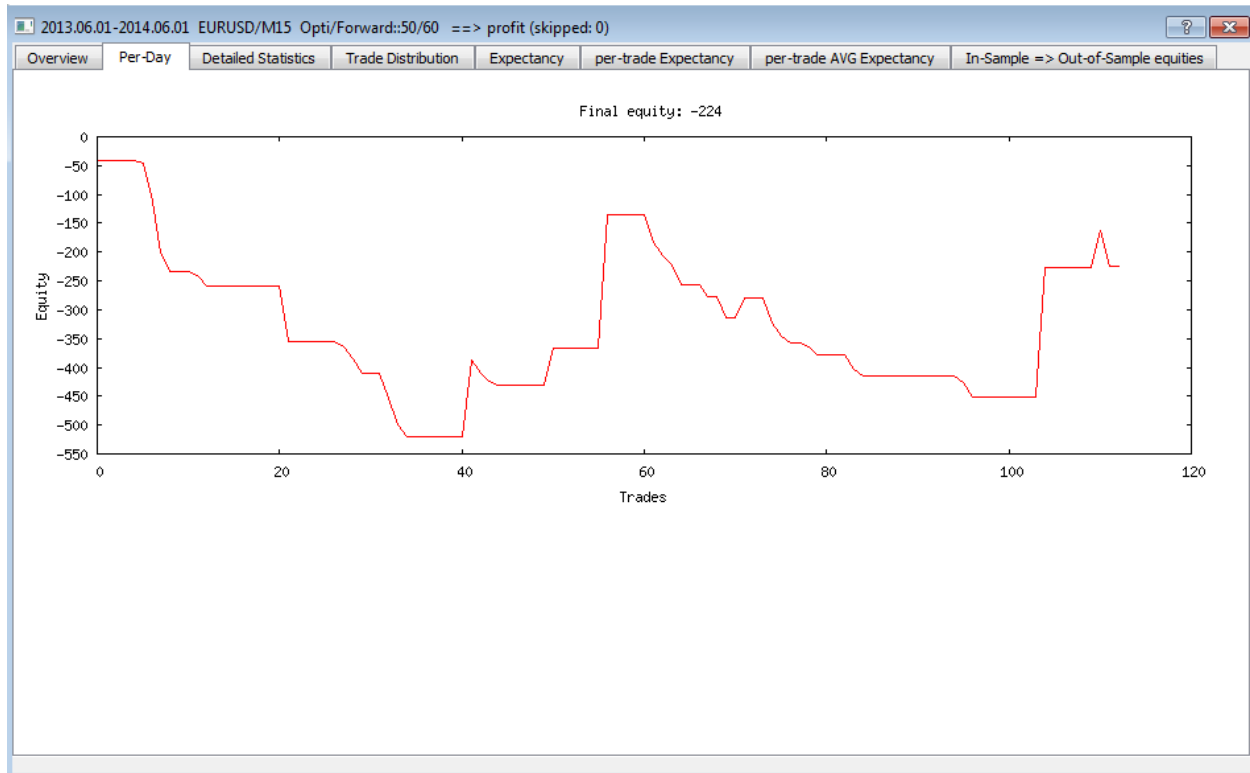
1. The equity and some core-statistics of the WFA. This, of course, only includes out of sample trades (from forward trading, no data from optimisation is in here!)



2. In a WFA, parameters change frequently, due to re-optimisation. So you can not just put a WFA-optimised EA into metatrader and watch it trade. Because of this, DATFRA can take all trades from a WFA and construct a dummy EA out of them, that trades exactly as the EA did during WFA. That way, you can watch the single trades, over all WFA windows, in Metatrader4. Use this button to do so!



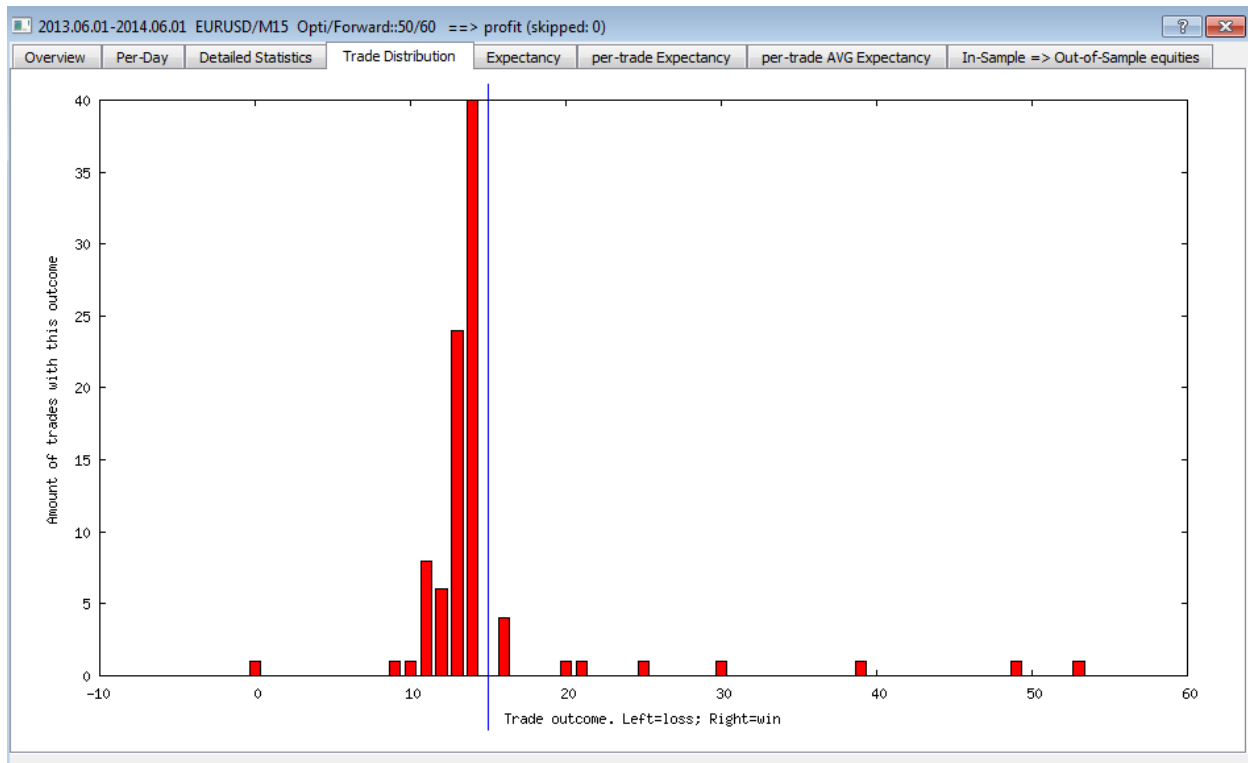
3. This is the same equity, but this time each point is not one trade (as in the equity you saw before), but each point is one trading day!



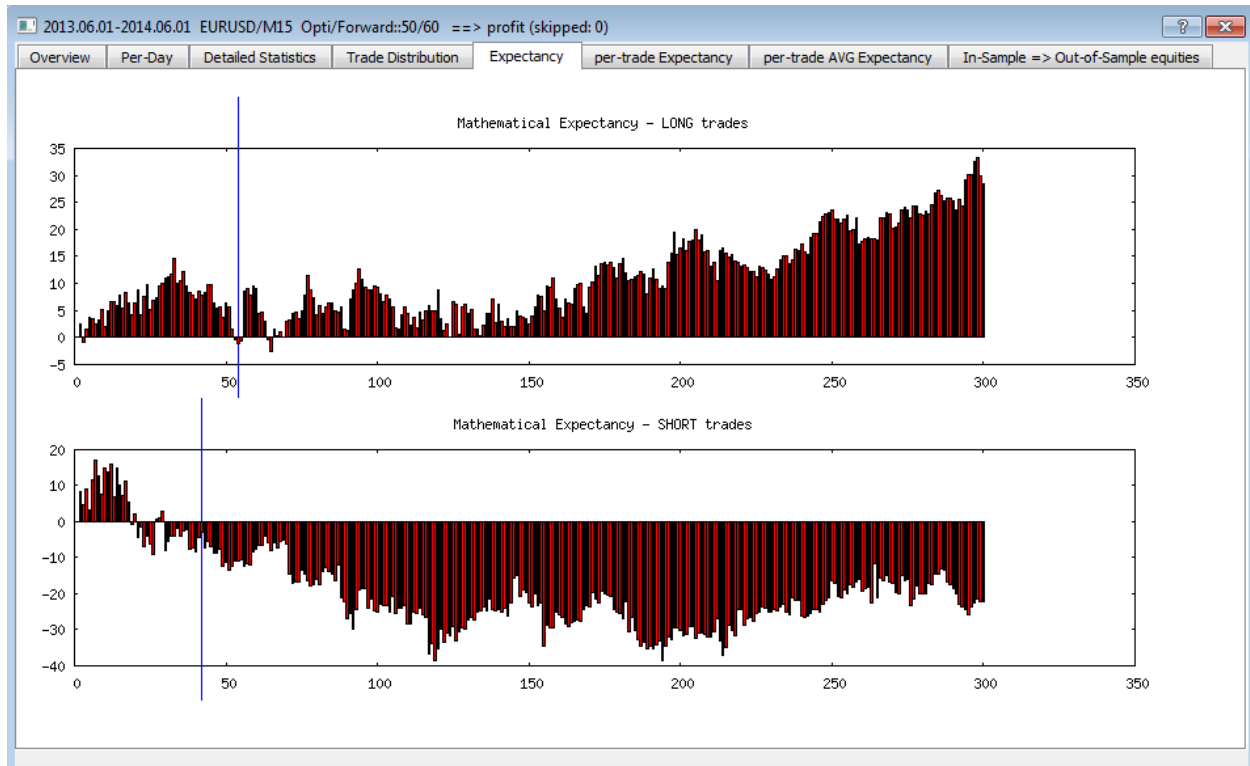
4. Statistics of your WFA

2013.06.01-2014.06.01 EURUSD/M15 Opti/Forward::50/60 ==> profit (skipped: 0)						
Overview	Per-Day	Detailed Statistics	Trade Distribution	Expectancy	per-trade Expectancy	per-trade AVG Expectancy
		Statistic	Overall	Long	Short	In-Sample => Out-of-Sample equities
		trades_monthly	24.8849	18.8493	13.7838	
		trades_total	92	41	51	
		hit_endofbacktest	0.0108898	0.0243902	0	
		hit_takeprofit	0	0	0	
		hit_stoploss	0	0	0	
		smoothness	0.092041	0.562402	0.0502547	
		DD_relative%	5.08189	3.21017	5.08189	
		DD_absolute	505.48	319.54	505.48	
		DD_avgLength(trades)	91	40	50	
		DD_maxLength(trades)	91	40	50	
		profit_yearly-yield-per-DD	27.0188	36.2298	24.4751	
		profit_yearly-yield	136.765	113.093	123.89	
		profit_monthly	-60.7919	-11.9712	-52.9189	
		profit	-224.93	-29.13	-195.8	
		profit_factor	0.799882	0.931804	0.718516	
		gross_win	897.82	398.02	499.8	

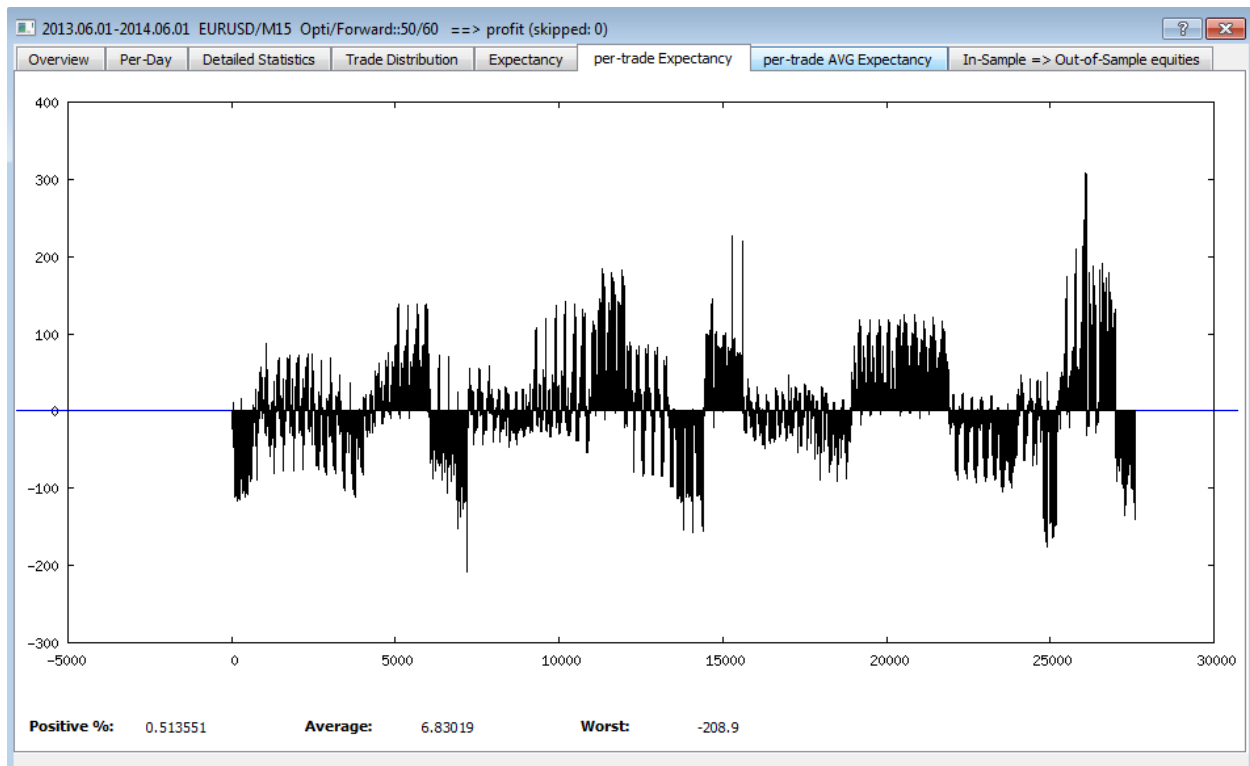
5. This shows how the outcomes of your trades are distributed. On the left side of the blue line are the losers, on the right the winners. The closer the bar is to the line, the smaller was the amount of the trade. The height of the bar indicates how often a given trade-outcome happened. So here we have a lot small losing trades and a few pretty high winners!



6. This shows you how far the price goes in your favour - or against you - after an average long/short entry. X-axis: How many bars after the entry this was measured. Y-axis: How many pips the price moved in average.



7. This shows you how far the price did go in your favour - or against you - after every single entry. Each line is one bar, and for each entry every single bar, until the trade is exited, is measured and displayed here.



8. This shows the same data as the last one - but this time each bar is the average price movement of one single trade.

