Kantu, a Parameterless, Price-action based trading system generator

Made by Daniel Fernandez

Contents

Introducing Kantu

Algorithmic system generation

During the past several years there has been a dramatic increase in the number of algorithmic participants in the world markets. This increase has led to an "algorithmic arms race" in which traders compete for the creation of more efficient and profitable trading implementations. Traders who tackle the creation of algorithmic strategies by manually designing new systems suffer from a tremendous disadvantage as their development capabilities are heavily surpassed by other traders who are capable of generating candidate system implementations in an automatic manner. By using software with data mining capabilities, a trader is able to generate thousands to millions of potential strategies while a trader manually developing a strategy might take years to find and code even a few algorithms with the same historical results.

Due to the above facts, the capability to design systems automatically becomes extremely important and therefore the creation of easy-to-use software that allows traders to get a deep insight into the system creation and analysis process becomes necessary. My objective with Kantu is to design a program that allows regular retail traders to access powerful data-mining capabilities, allowing anyone to quickly evaluate very large amounts of strategies and get insights into new potential edges and trading methods.

What Kantu is

Kantu is a white box Parameterless, price-action based system generator that evaluates thousands of different potential trading strategies on a given amount of historical data to obtain results that match some pre-defined user filters. All the strategies created by Kantu lack any parameters that determine either the entry or exit logic (when no SL/TP is used) and therefore they are not subject to many of the problems inherent to parameter optimizations within indicator based or other parameter containing trading strategies. Kantu systems act only on already closed bars, meaning that Kantu systems are easy to simulate accurately. Systems generated with Kantu can be easily interpreted by the trader as the price patterns are usually simple and transparent. **The Kantu program is effectively a white-box as the user has complete knowledge of the trading logic used by the algorithms that are ultimately created**. Kantu can help you do the following things:

- Attempt to find historically profitable price-action based patterns across any instrument on any time frame across any selected period.
- Evaluate the historical performance of a given system generation/selection methodology.
- Study the relationships between in and out of sample performance
- Study the historical effect of different filters in systems generation on out-of-sample performance
- Get a coarse framework which can be later improved manually by a trader

What Kantu is not

Kantu is a tool that allows you to find system implementations that satisfy some historical performance requirement. **However Kantu is in no way a holy grail builder** as there is no way to guarantee the future profitability of the strategies developed by the software. No trading software can provide a guarantee of future performance for any strategy and therefore it would be misleading to assume that the creation of historically profitable systems will inevitably lead to profitable future trading. **Kantu will not guarantee your profitability in trading** and in fact usage of the tool without a good understanding of algorithmic trading and the potential issues behind automatic trading system generation (such as data-mining bias, data-snooping bias) are bound to lead to bad results. **Your ability to be profitable with Kantu generated strategies will depend on how you use the software and how you select the trading strategies created by it**. Therefore please bear this in mind when using Kantu:

- Strategies generated with Kantu are not guaranteed in any way to be profitable in the future
- A positive historical performance is no guarantee of future profitability
- Simulated performance might be inaccurate when the simulated time-frame is too small (this depends on the instrument). In general staying above the 30 min time frame is advised.
- Strategies are mostly guaranteed to fail if curve-fitting and data-mining bias measures are not taken into account.
- **Kantu is no holy grail creator.**

A summary of Kantu's capabilities

Here is a summary of Kantu's main features:

- The program is available on Windows and Linux.
- **Multi-core support** (choose the number of CPU cores you want to use).
- Generate Parameterless, price-action based strategies on any Forex or non-Forex instrument on any time frame. Note that shorter time frames take longer to simulate due to the larger amount of bars.
- Save custom instrument information including spread, commission, contract size, etc.
- **Multi-instrument simulations.** Find patterns that fit some historical profitability criteria across several different instruments.
- Choose which variables to use in system creation. Use only OHLC values or implement custom variables.
- Implement hourly filters or daily filters within simulations
- Control whether or not to enforce system symmetry.
- Obtain mathematical expectancy measurements for each strategy's signals.
- Test any single system you want by defining rules and filters manually.
- Control pattern complexity by selecting the maximum number of rules and shift to use.
- Select from a wide variety of statistics to filter generated trading results (profits, max drawdown, winning percentage, profit factor, ulcer index, Pearson linear regression coefficient, kurtosis, skewness, etc)
- Sort generated systems by any of the statistical criteria included within the program (ascending or descending)

- Create a custom linear filter that combines any of the statistics in any proportion. You can create a custom mathematical formula combining the filters as well.
- Choose in/out of sample periods. Profitability for the out-of-sample period is automatically calculated on each run.
- Generate full historical graphs for each generated strategy on a single click. Linear regressions as well as in/out of sample division lines are shown.
- Export to pseudo code
- Export to MQL4 code with one click.
- Manually set the spread level used on simulation
- Volatility based money management is automatically incorporated into the strategies.

Understanding Kantu Basics

Defining price-action based patterns

Kantu generates systems that trade using price-action based patterns. These patterns are defined through simple comparative relationships between different candles, allowing for the generation of patterns without the introduction of any parameters to control the opening or closing of trading positions. Kantu defines patterns by generating random rule combinations of Open/High/Low/Close variables as well as Body and Range features. The OHLC values for an instrument can be compared to any different OHLC value while Range and Body values are only compared between each other. In addition the program can also use any additional inputs in the OHLC file which will be compared to any other values except the Body and Range.

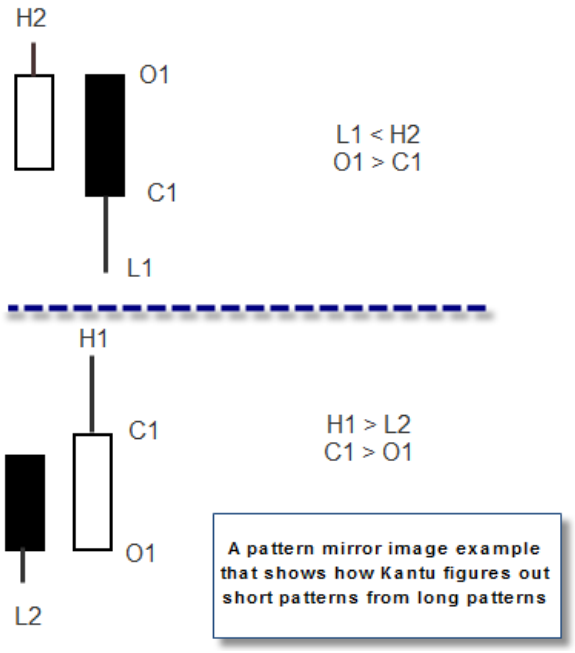
As an example Kantu might generate a 4 rule pattern defined in the following manner:

- Close[2] > Open[2]
- High[2] > Low[5]
- Open[1] > Close[5]
- Body[3] > Body[7]

The program would use this as the long entry logic for the defined pattern. The logic to enter shorts would be the direct inverse of this pattern in a way that is perfectly symmetric (meaning that if you looked at the graph upside down you would enter a trade in the opposite direction).

- Close[2] < Open[2]
- Low[2] < High[5]
- Open[1] < Close[5]
- Body[3] > Body[7]

The following image better explains how the inversion of patterns works in order to figure out the short entering rules after the main patterns (which are used for longs) are generated. Note that Kantu always uses the exact same pattern for long/short entries in order to make sure that the trading strategy generated is symmetric in nature (from a logic perspective). It is also worth noting that Kantu opens a new long trade and closes any existing short trades when a long signal is detected, and vice versa for short signals.



Kantu also contains code that prevents the generation of patterns that do not make sense (for example patterns where there are contradictory rules) as well as code that prevents the creation of patterns that have already been added to the final results. It is however worth noting that this is not a guarantee of a lack of correlation between the generated strategies as strategies can become very similar if they are only different due to rules that have little effect. It is always advisable to evaluate different strategies created on the same symbol to determine how close their trades actually are to one another.

Money Management

Kantu implements a money management system based on adjusting lot sizes against the instrument's volatility. This technique has proved to give better and smoother results for strategies in our experience at Asirikuy.com. The lot size is adjusted so that a movement of 200% of the 20 period ATR will give you a loss of 1% of the account's value. Kantu simulations do not implement any compounding so this value always matches 1% of the initial account's size (which is 100,000K USD). Please note that for time frames lower than the daily the ATR value is adjusted in a manner which is proportional to the difference between this time frame and the daily. For example on a 1H back-test the system will adjust lot sizes against 200% of the 20 period hourly ATR multiplied by 24, on a 30 min chart lot sizes would be adjusted against 200% of the 20 period 30 min ATR multiplied by 48. Please also note that these same ATR multipliers are used when SL or TP values are implemented. Additionally when the system has an active SL the lot size will be adjusted against the system's SL value (so that a hit of the SL equals a loss of approximately 1% of the initial account balance). Options for additional money management strategies might be implemented going forward to expand the possibilities generated by Kantu.

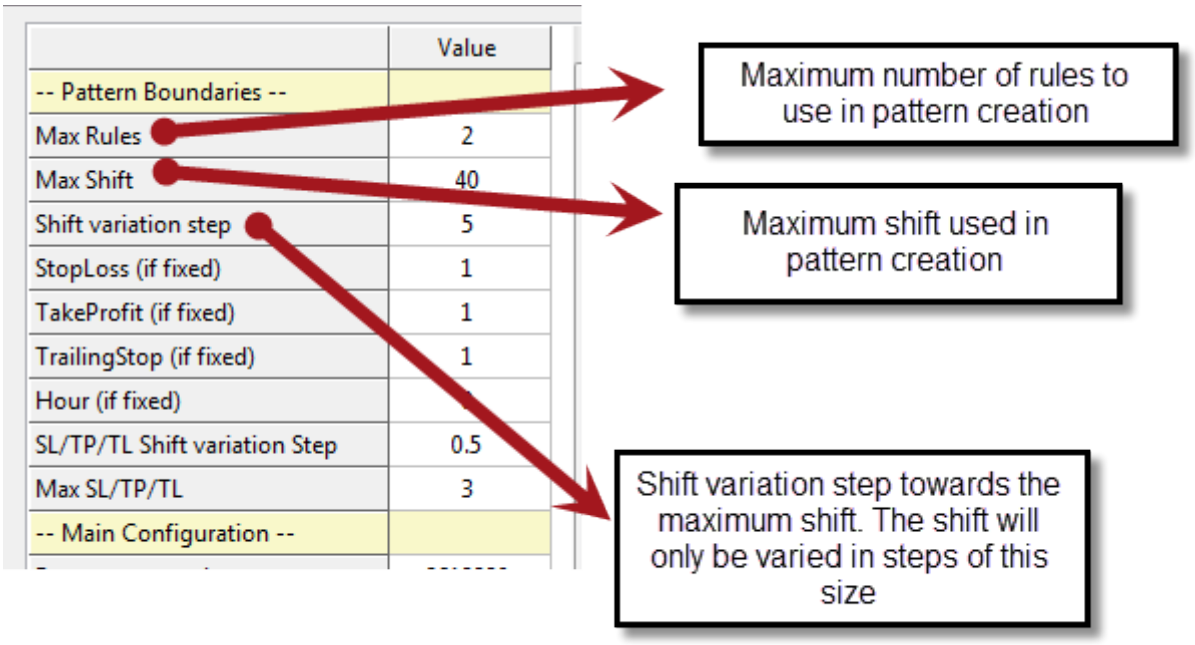
It is also worth noting that the money management is not parameter-less as the adaptation against volatility requires a gauging of the movements of the pair within a given number of past periods. However the choice of ATR period doesn't affect results very much as it does not have any influence on

when positions are opened or closed. In general the results of the system will converge at large ATR values, reason why it is not advisable to use low values for this calculation. However the user can modify the way in which the money management is carried out once the system is exported to either MT4 or F4.

When exporting systems to MQL4 code the money management assumes that the account trades in USD, tests of different risk levels will be necessary to ascertain proper equivalent risk levels for other currency pairs. It is advisable for the user to perform back-tests that verify the correct function of the system before trading the system on a live or demo account. When systems are exported to F4 the framework automatically corrects risk for any combination of pair and deposit currencies.

Configuring pattern generation

Now that we know how Kantu builds trading patterns we can now understand how the configuration affects the patterns being generated. Kantu allows you to define the maximum pattern complexity by specifying the maximum rules per pattern and the maximum shift that you would like to take into account. Increasing any of these values will increase the logic space (the amount of potential strategies) meaning that simulations might take longer to find desirable results, while you might also find results that were not possible on less complex patterns.



The default values (2 max rules and 50 max shift) lead to very simple (what could be called elegant) price patterns, but certainly on some instruments patterns with better historical results might be achieved by increasing pattern complexity. In general it is advisable to attempt to increase the shift first (to a value between 100 and 300) and then attempt to increase the rules per candle. The rule number has the highest effect on the parameter space so it's advisable to keep this number low in order to ensure that pattern searching times will not become extremely long.

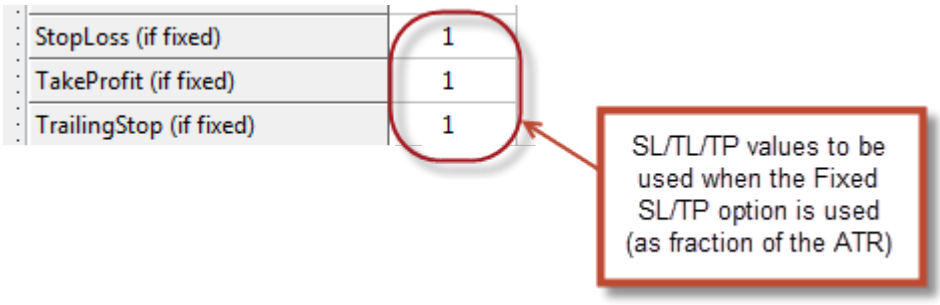
There is also a "Fix complexity" option that allows you to test a given specific pattern composition without looking for simpler patterns as well. For example if you set the "Max rules per candle" to 5 and then select the "Fix Complexity" option, the program will only look for patterns with 5 rules per pattern (it won't look for patterns with less than 5 rules).

The "Calculate system number" button will also calculate the number of possible systems for the logic space selected. For example when running a generation method with 2 rules, a max shift of 40, a shift step of 5, a max SL/TP/TL of 3, with a step of 0.5 and option "Use Stop-Loss" checked, the number of total systems will be 2'618,880. Note that Kantu generates systems randomly and therefore generating 2'618,880 does not guarantee that all the systems within the logic space would have been tested.

Configuring the System

In Kantu you can select different system types depending on the amount of freedom you want your system to have. Systems are parameter-less by default and contain only one price action based pattern that determines both entries and exits. This is an always-in-the-market type of approach that implies that you will always be trading after you enter your first position. By default Kantu uses no stop-loss or take-profit values but if you want these values can be activated by checking the appropriate box. If you want to avoid an always-in-the-market approach the best solution is probably to include an additional Parameterless closing pattern – which is different from the entry pattern – that will allow you to have periods where your system does not have any open trades. Here is a list of the available options and their effect:

- **Use Stop-Loss:** Enables the use of an ATR adjusted stop-loss within the trading strategies. This value will always be between 50 and 250% of the ATR.
- **Use Take-Profit:** Enables the use of an ATR adjusted take-profit within the generated strategies. This value will always be between 50 and 250% of the ATR.
- **Use AFM-TL** (Always on Favorable Move – Trailing Stop): Enables the use of a trailing stop that uses no additional parameters. This trailing stop simply moves the stop-loss on favorable moves so that it is always the same distance away from the most favorable candle open taken. This is a non-retracing trailing-stop, meaning that the stop-loss is never moved into a more unfavorable position.
- **Use Fixed SL/TP:** Fixes the SL/TP/TL values to those defined within the simulation options.



- **Use Hour Filter**: Only enter trades on a particular trading hour.

- **Use Fixed Hour**: Forces systems to use the hour defined within the simulation options (Hour (if fixed option)).
- **Use Day Filter**: Only enter trades on a particular day of the week.
- **LR Through origin:** This option makes linear regressions go through the origin of the balance curves. This means that systems need to have stable growth from their start, since they are punished heavily by the regression model from deviations.
- **Asymmetry:** Generates systems that trade long-only or short-only positions. Trades are still exited on reverse signals but no opposite signal is opened. This option is useful for markets where there is a specific long term bias (as equities). When this option is clicked a dialogue appears asking what type of system to generate (short only or long only).

Configuring Simulations

Kantu simulates systems by using a fixed trading cost model that is appropriate for trading on the upper time frames (> 30min). Using Kantu on the lower time frames might not be advisable if you are not experienced with Forex simulations and the problems that this might generate when proper Bid/Ask feeds are not taken into account. Additionally Kantu allows you to specify random slippage variables that will allow you to simulate the effect of none ideal execution on your trading strategy. You can choose how much will be the maximum allowed slippage as well as how many trades will suffer from a random slippage distortion. In Kantu slippage is assumed to always be negative as this is most often the case in Forex trading.

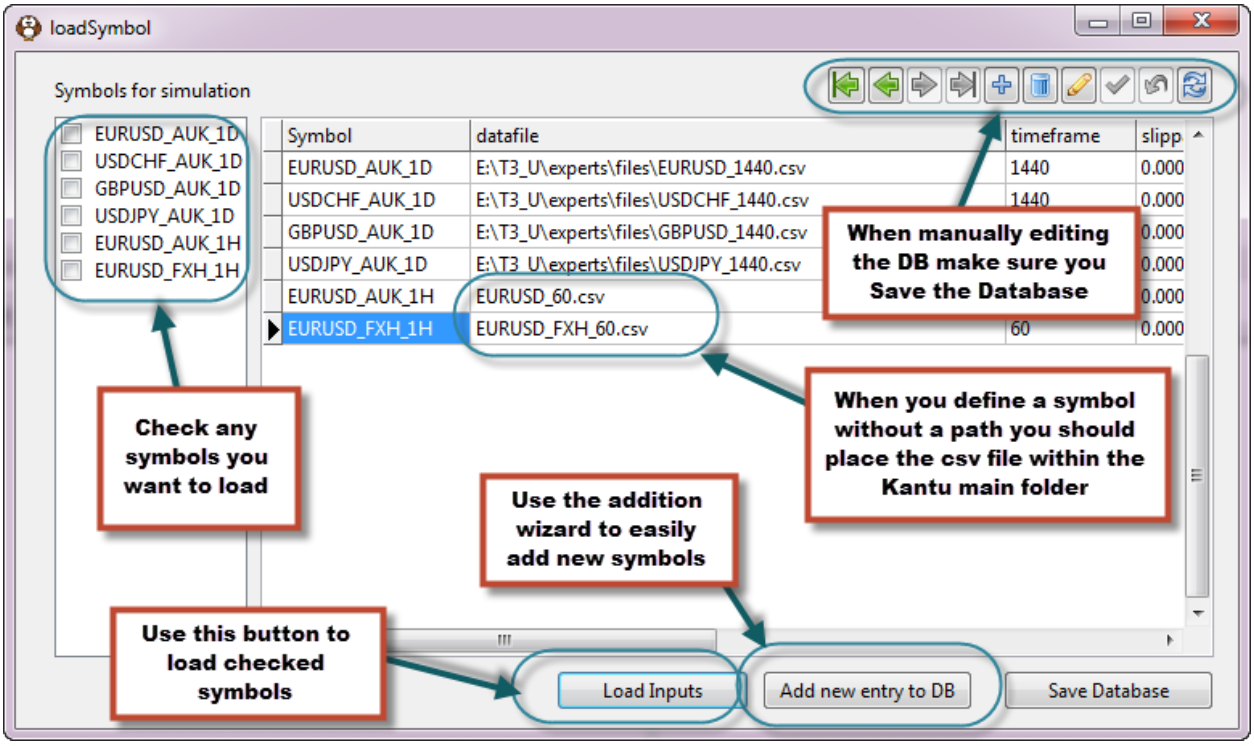
Loading historical data

Kantu requires data to be loaded in a comma delimited csv format. The data should be formatted in the following way (this is the format that the MT4 platform creates when you export a currency pair's data):

yyyy.mm.dd, hh:mm, Open, High, Low, Close, Additional Input 1, …. Additional Input X

The data should contain the date, open, high, low and close information. In order to have results that are as reliable as possible it is also recommended that the historical data used for simulations and the data that will be used for live trading have exactly the same composition (meaning the same weekly starting/ending times, DST shifts, etc). **The data should also go from the oldest date to the newest date in order for the arrays to be populated correctly.**

Once you generate an appropriately formatted csv file (exporting your data from the MT4 history center is recommended) you can load the symbol by adding an entry into the symbols database which is located in your symbol/ folder inside your Kantu installation folder. Entries into the database can be added by manually editing the txt files or they can be added by using the DB manager within Kantu.



The symbol field of the database can be defined in any way you desire while the other fields must contain the appropriate values for the instrument you want to load. It is especially important to enter the absolute path to the history csv file within the datafile filed and also to enter the timeframe (in minutes) under the time frame column.

```
Symbol;datafile;timeframe;slippage;spread;contractSize;commission;isVolume;pointConversion;roundLots;MinStop
"EURUSD_AUK_1D";"EURUSD_1440.csv";"1440";"0.0005";"0.0003";"10";"0";"1";"10000";"2";"0.0015"
"USDCHF_AUK_1D";"USDCHF_1440.csv";"1440";"0.0005";"0.0003";"10";"0";"1";"10000";"2";"0.0015"
"GBPUSD_AUK_1D";"GBPUSD_1440.csv";"1440";"0.0005";"0.0003";"10";"0";"0";"10000";"2";"0.0015"
"USDJPY_AUK_1D";"USDJPY_1440.csv";"1440";"0.0005";"0.03";"10";"0";"0";"100";"2";"0.15"
"EURUSD_AUK_1H";"EURUSD_60.csv";"60";"0.0005";"0.0003";"10";"0";"1";"10000";"2";"0.0015"
"GBPUSD_AUK_1H";"GBPUSD_60.csv";"60";"0.0005";"0.0003";"10";"0";"1";"10000";"2";"0.0015"
"USDJPY_AUK_1H";"USDJPY_60.csv";"60";"0.0005";"0.03";"10";"0";"0";"100";"2";"0.15"
```

The above image shows you an example of the database information contained within the symbols.txt file. The instrument database is a simple semicolon separated csv file which allows for easy editing of values in case you face any problems with editing on the DB editor inside the program. Here is a description of the meaning for each one of the different fields:

- **Symbol:** Identifier of the instrument you are loading. If you are loading several time frames for the same instrument it is better to give them different symbol names (for example EURUSD_1D and EURUSD_1H for a daily and hourly EURUSD feed)
- **Datafile:** Absolute path where the csv file containing the data for this instrument is found.
- Timeframe: specify the timeframe for the data you are loading in minutes.
- **Slippage:** Maximum slippage to be added to each trade (according to the slippage probability specified in the options)

- Spread: The spread charge in absolute price units for this instrument
- **contractSize:** The cost of trading per pip at the standard contract size (1 lot). In the case of Forex instruments this is usually 10 USD for standard accounts.
- **isVolume:** Whether the data contains a volume column.
- **pointConversion:** Multiplication that must be carried out to change the price value of the symbol to pips. For example in the EUR/USD one pip is 0.0001 so the pointConversion is 10,000. On the USD/JPY a pip is 0.01 so the point conversion is 100. Another example is the DAX where one pip is 1 and therefore the point conversion is 1.
- **roundLots:** To how many decimal places you want to round lots when running simulations. In Forex this is usually 0.01 while in futures and stocks it is usually 0 (no fractional lot usage is permitted).
- **MinStop:** The minimum stop and limit distance for this symbol in absolute price units. For example a value of 0.0015 on the EUR/USD implies that your broker will not let you set a stop that is less than 15 pips away from current price.

There may be some limitations in the amount of characters you can add when using the DB editor. To get around this problem you can either edit the txt database file directly or you can use the “Add new entry to DB” button, which allows you to add a new instrument through an interactive wizard that allows you to easily enter all the data, browse to the appropriate csv file (with automatic definition of the path), etc. The image above shows you where this button is located.

When loading data please also remember to check the checkbox for the instrument you want to load and press the “Load Inputs” button.

Using Kantu

Simulation Types

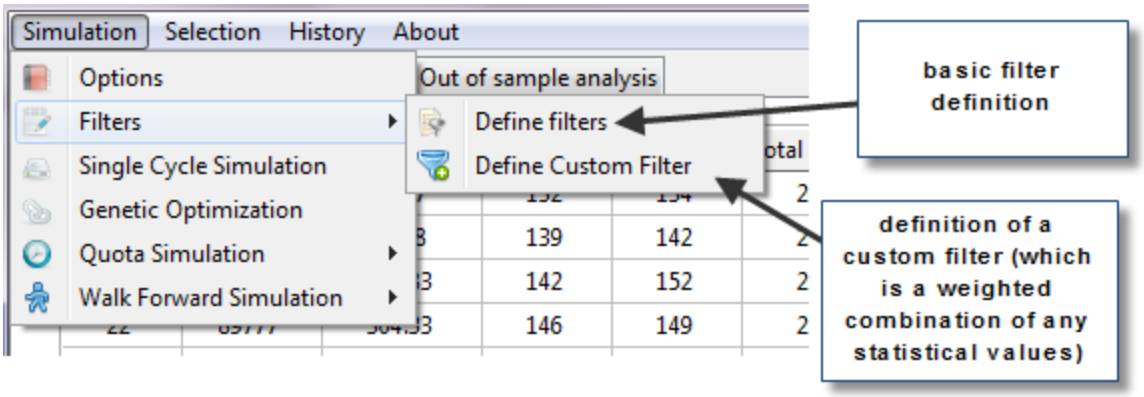
Kantu offers a wide variety of possible simulations that will hopefully allow you to explore algorithmic system generation as well as the development of system generation methodologies in depth. Here is a list of the types of simulations available:

- **Find systems (main symbol):** This process generates systems until the requested number of systems is found. This means that Kantu will generate strategies until it has gathered a given number of systems that pass the input filters. Depending on how astringent the filters are Kantu might need to generate thousands or millions of strategies to satisfy the requested strategy quota. The OS result values for this run type will be from the end of your in-sample period (ending date) to the end of your data. If no strategies with the desired characteristics are present the process may go on forever. Remember that there is no guarantee that a strategy with certain characteristics will ever be found.
- **Find systems (multiple symbol):** Through this simulation Kantu will perform a normal fixed quota simulation but will only add results that fulfill the specified filtering criteria across all the loaded trading instruments. This allows you to find price patterns that give historically profitable results across many different symbols.

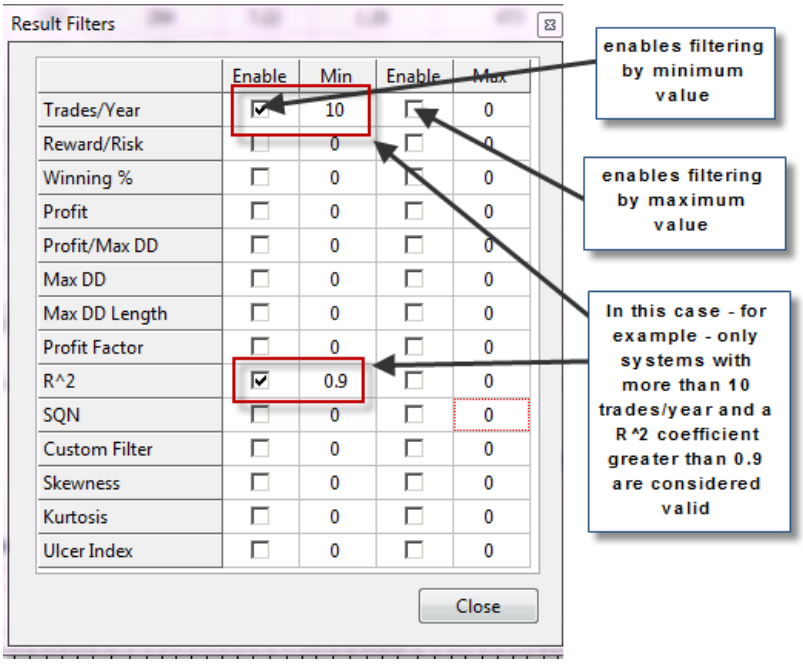
Note on multi-core support: Kantu supports multi-core usage across all simulation types. Simply set the number of cores you wish to use within the “Number of cores” option. The default is 3 cores.

Using Filters

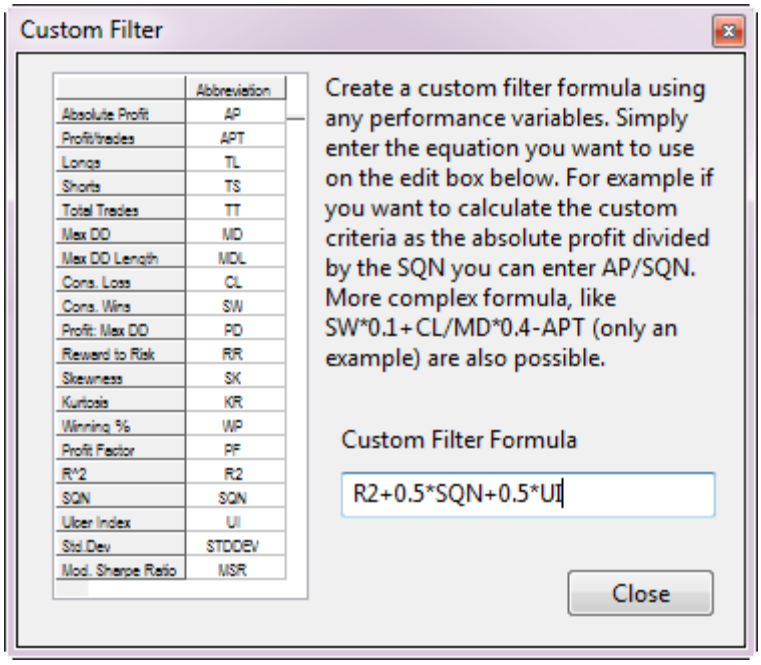
Trading strategies generated using Kantu can be filtered using a variety of custom criteria. In general the strategies are automatically filtered for symmetry and profitability, meaning that only profitable and symmetric systems (difference between longs and shorts is less than 50% of total trades) are considered valid, but other filters can be applied in order to obtain strategies that satisfy some given statistical thresholds. Basic filters for statistical values can be defined through the “Define filters” menu item under the Filters sub-menu while a custom filter (weighted combination of any filters) can be defined through the “Define Custom Filter Item”



The next image shows you the "Results Filter" dialogue that allows you to select minimum and maximum values for any of the statistics calculated by the program. With this you can select strategies that satisfy historical threshold for any of these statistics. The example below has a filter that will filter out all strategies that have less than 10 trades per year or a squared Pearson coefficient below 0.9. A strategy will only be considered valid if it satisfies all the filters that you have selected. The more complicated your filters, the more systems that will be generated in order to satisfy your selection criteria.

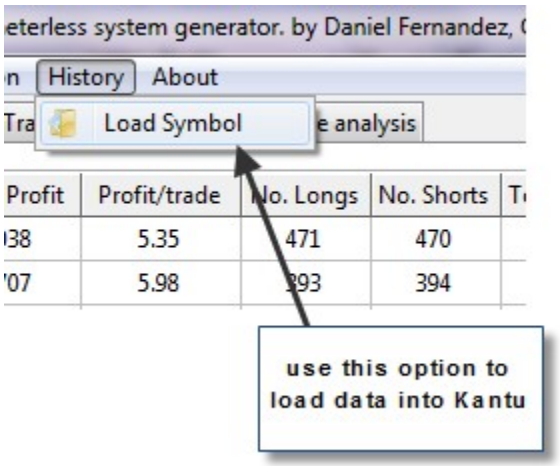


The custom filter form, showed below, allows you to create a custom filtering statistic that is simply a linear combination of any of the other available values. For example I can create a filter – as shown below – that is simply the R^2 plus half of the SQN minus half of the Ulcer Index. This will generate a custom statistic that I can use to filter my generated results. This is useful as logistic regressions or other models might generate possible clues into the historical relationship between in and out of sample values that can be expressed as a custom statistic of this form.



Basic system generation run

The most basic type of run in Kantu is the creation of a given fixed number of trading systems that pass your requested filters. In this mode the program simply generates a given number of requested strategies and only those that pass the profitability, symmetry and custom filters are showed as results. In order to run a basic system generation cycle you should first generate appropriate historical data csv files (as mentioned in the previous section). Once you generate this information you can then load the history file by using the "Load Symbol" option within the History menu (as showed below). This will open up the symbol DB manager which will allow you to edit the symbol database and load any symbols you want to test. Closing the symbols database loads any requested symbols.

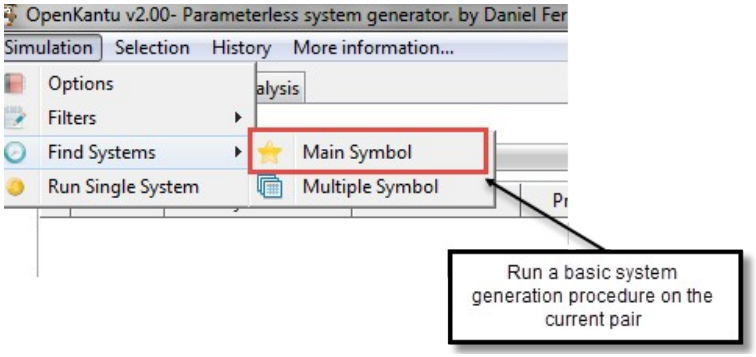


Kantu also filters your data in order to ensure that there are no corrupt dates within the array. If there are any inconsistencies in your data (such as a future date that is older than a past date), Kantu will show a message with the corrupted date information so that you can further debug your data.

Once you have loaded historical data you can now proceed to configure your run by going to the options menu as showed above. This will open up the options dialogue where you can select the main aspects of the system generation process and simulation. On this dialogue you can select the complexity of the created price patterns as well as the trading costs and the periods used for the simulations. Here is a summary of the options available and what they mean:

- **Max Rules**: Maximum rules to be used for each created pattern.
- **Max Shift**: Maximum shift to be used within each created pattern.
- **Shift Variation Step**: Control the step variation used to create systems between 1 and the Max Shift used. For example if you select a max shift of 100 and a shift variation step of 5 system with shifts in 5 multiples (5, 10, 15 … 100) will be created. This is useful to limit the size of the logic space.
- **StopLoss (if fixed)**: Value of the SL (fraction of the ATR) used when the "Use Fixed SL/TP" option is checked.
- **TakeProfit (if fixed)**: Value of the TP (fraction of the ATR) used when the "Use Fixed SL/TP" option is checked.
- **TrailingStop (if fixed)**: Value of the TL (fraction of the ATR) used when the "Use Fixed SL/TP" option is checked.
- **Hour (if fixed)**: Value of the trading hour filter used when the "Use Fixed Hour" option is checked.
- **SL/TP/TL Shift variation step**: Controls the variation step of the SL/TP/TL values. For example if you set the maximum value to 3 and the step to 0.2 the SL/TP/TL values will be generated only in 0.2 increments (0.7, 0.9, 1.1…3.0). Note that no SL/TP/TL value is every assigned below 0.5 to maintain simulation accuracy.
- **Max SL/TP/TP**: Maximum possible value of the SL/TP/TL values as a fraction of the ATR.
- **Number of cores**: The number of processor cores you wish to use.
- **Number of requested systems**: This option controls the number of systems that are generated on Fixed Quota and Walk Forward simulations.

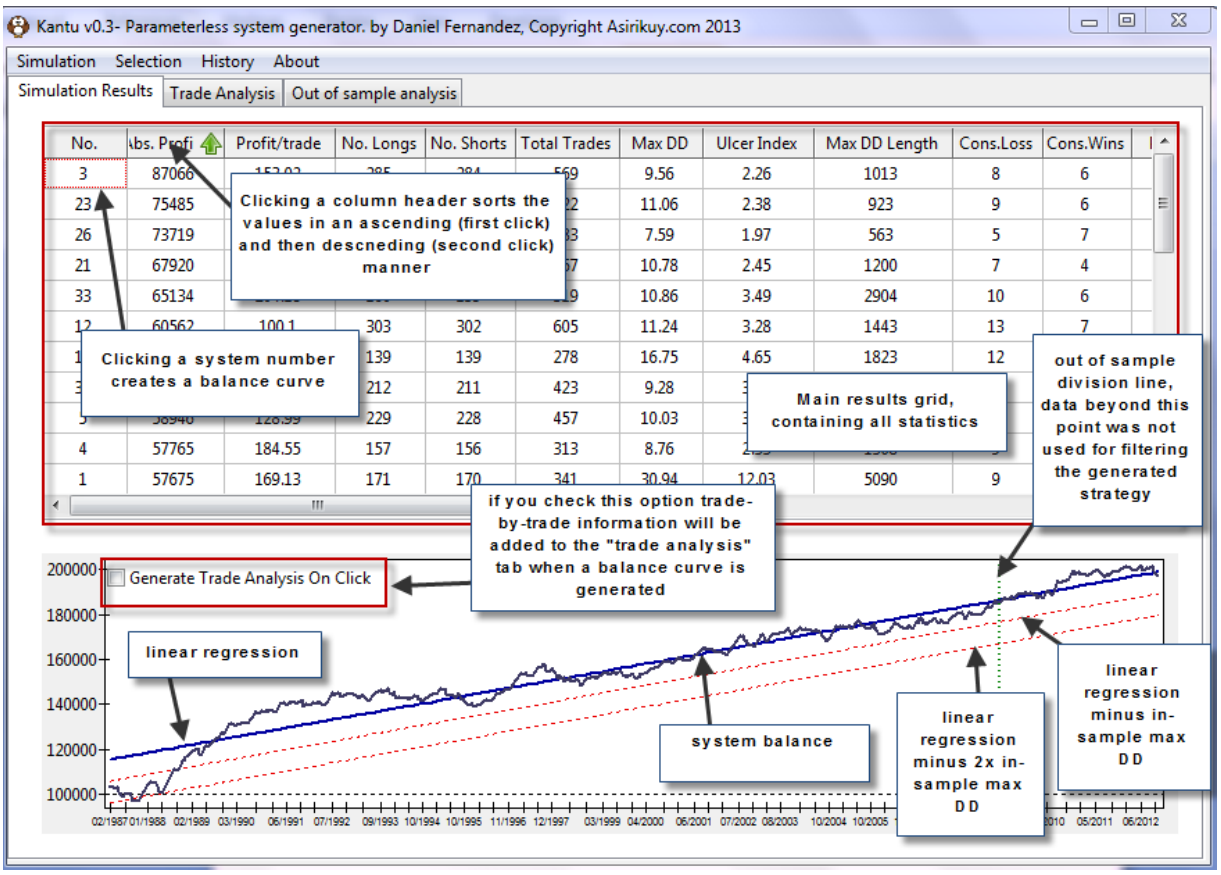
Once you have configured the number of systems you want to generate as well as in-sample period length (starting and ending dates), you can now run the simulation by choosing the single cycle option from the Simulations menu.



A progress bar will appear showing you how many systems have been generated from the number of systems requested. Note that only systems that pass the filters will be added to your results.

Using the main results tab

Results that have passed the profitability, symmetry and custom filters are then copied onto the results grid. This grid shows you all the statistical results for your system and allows you to create graphs with a single click. The grid also includes out of sample total profit and out of sample per trade profit information which you can use to evaluate the performance of the strategies under unseen market conditions.



You can sort the statistics in any way you want by clicking the header for each statistic and you can also generate a balance graph for each system by clicking on the number for the system on the “No.” column. Additionally if you check the “Generate trade analysis on-click” checkbox you can actually generate a trade-by-trade analysis of the strategy under the “Trade Analysis” tab.

If you want you can also export the results of the grid onto a csv file by right clicking on the spreadsheet and selecting “Save contents to CSV”, this process can also be performed on the “Trade Analysis” grid and on the balance graph (where you can save an image instead). This will allow you to export your data to perform more complicated statistical analysis on any software of your choice.

Summary of Result Grid Statistics

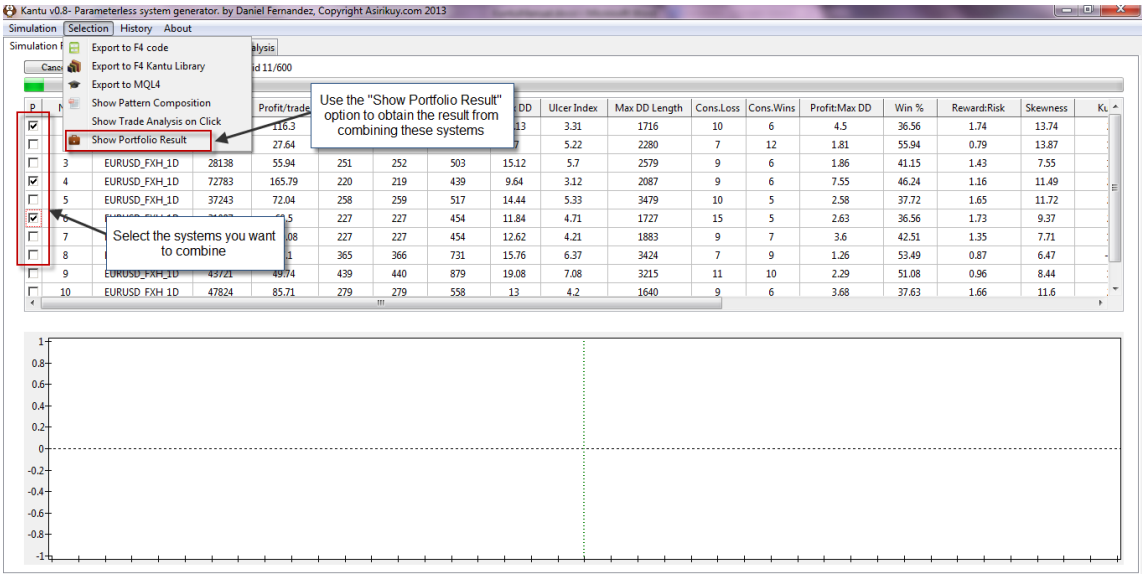
The result grid provides many useful statistics to help you analyze generated trading strategies. Here are the definitions for the different values you find there:

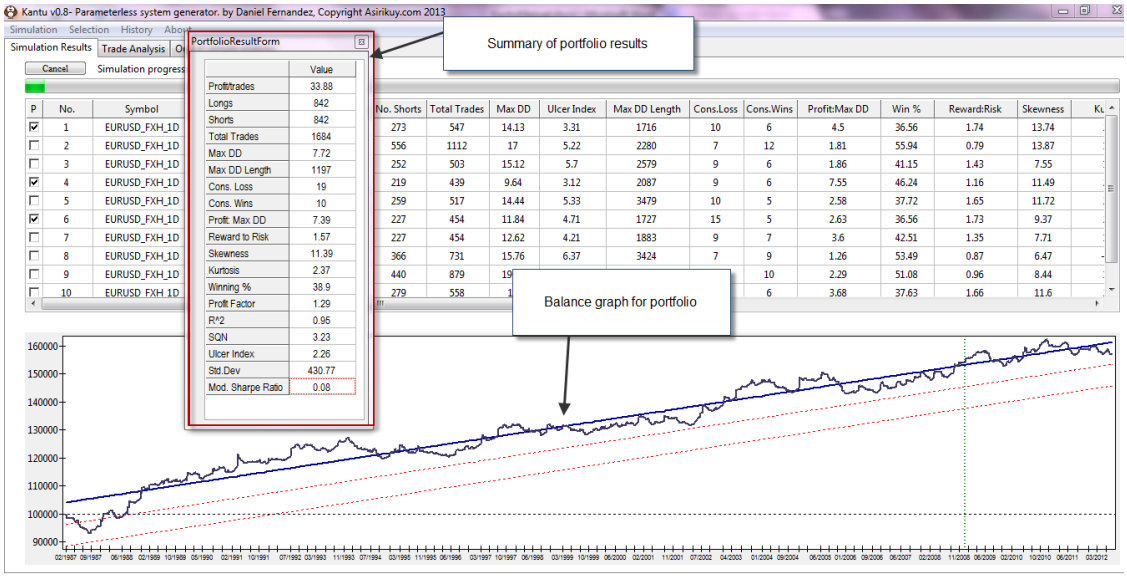
- **P:** Check this box if you want to select this system for a portfolio analysis.
- **Number**: Trade number identifier. It means that this was the nth system generated.
- **Abs.Profit:** Absolute Profit for the strategy defined as the difference between the final and initial balance values.
- **Profit/trade**: The absolute profit divided by the total number of trades.
- **Profit Longs**: The total amount from the absolute profit belonging to long positions.
- **Profit Shorts**: The total amount from the absolute profit belonging to short positions.
- **No. Longs**: The total amount of long trades.
- **No. Shorts**: The total amount of short trades.
- **Total Trades**: The total amount of trades taken by the strategy.
- **MaxDD**: Maximum drawdown – maximum distance between a balance high and balance low – defined as a percentage of the initial balance.
- **Ulcer Index:** Implemented as defined here.
- **Max DD Length:** Maximum time (in days) between two consecutive equity highs.
- **Cons.Loss:** Maximum number of consecutive losing trades.
- **Cons.Wins**: Maximum number of consecutive winning trades.
- **Profit:MaxDD**: The absolute profit per year divided by the maximum drawdown.
- **Win %:** Percentage of trades that were profitable from the total trades.
- **Reward:Risk:** Average reward to risk ratio.
- **Skewness:** The skewness from the distribution of returns of the strategy (as defined here).
- **Kurtosis**: The kurtosis from the distribution of returns of the strategy (as defined here).
- **PF:** Profit factor defines as absolute profit over absolute loss.
- **Std.Deviation:** The standard deviation from the distribution of returns of the strategy.
- **Std.Dev Breach:** The maximum multiple of the standard deviation of linear regression residuals reached by the equity below the linear regression line.
- **Total ME:** Sum of all mathematical expectancy values across the number of bars defined by the user within the Simulation Options dialog.
- **Ideal R:** A statistic calculated based on multiple linear regressions to truly check how linear the trading strategy is and how well it approaches the ideal case. Go here to learn more.
- **R^2:** Linear regression correlation coefficient.
- **SQN:** System quality number as defined by Van Tharp. See definition here.
- **Mod. Sharpe Ratio:** Average trade profit over trade standard deviation.
- **Custom Criteria:** Result from the custom criteria equation as defined by the user.
- **Bars Out:** Number of bars that the system remained with no open positions within the simulation.
- **OSP:** Out-of-sample absolute profit.
- **OSP/trade:** Out-of-sample absolute profit divided by the total number of trades within the out-of-sample period.
- **Lowest Lag**: Lowest shift within the system's rule set. This gives you an idea about whether the system uses newer or older market information.

Note that when you save the results from the results grid as csv (right click the grid, Save as CSV) the csv will contain all additional statistics for the OS period (like the OS winning percentage, OS SQN, etc). You can also hide/show columns by right clicking on the grid and selecting columns from the Hide/Show columns menu. Your selections will be saved the next time you open Kantu.

Getting statistics for a portfolio

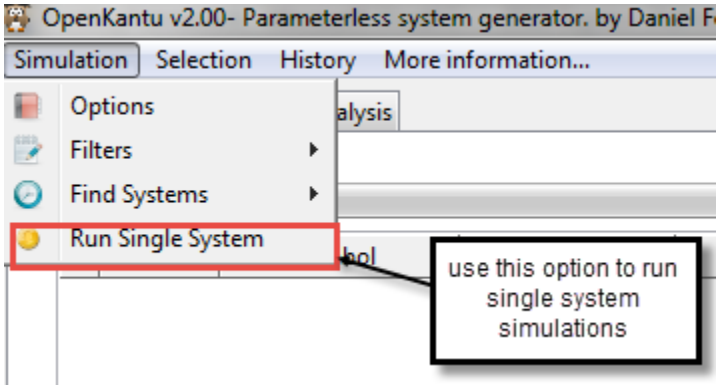
When you run a simulation you can obtain the results of a portfolio that combines several of your resulting systems by using the “Show Portfolio Result” option within the selection dialog. To do this simply select the systems you want to combine by using the “P” checkbox column and then go to the appropriate menu option. The following images explain the process. After using this option the program automatically merges the results for the selected strategies and displays the portfolio balance chart and statistical result summary.





Using the single system simulator

When using Kantu you will sometimes want to test a given price action based model manually. In order to do so go to the main menu and select the "Run single System" option. After doing this a dialog will pop-up where you will be able to manually create a given price-action based strategy.



On the "Run a custom single system" dialog you can define all the price pattern rules for the strategy you want to test. Simply add all the rules by using the input selector and the shift edit boxes. Please remember that you only add the rules for the long entry part of the strategy and the short rules are created automatically by Kantu using Kantu's definition of symmetry (*vide supra*). Note that Body and Range rules will be the same for long and short entry criteria as these rules do not have inverses (for example if you add a rule such as Body[1]>Body[2] it will be the same for longs and shorts). When you have already defined everything you need you can then execute the strategy by using the "Run Single System" button. The strategy result will appear within the results grid as when running a regular simulation. You can then export this system or analyze it as you normally do using Kantu's other tools.

When creating rules for single systems it is also important to create rule sets for which no contradictory signals are possible. For example a system with a rule Open[1] > High[5] will have an inverse rule Open[1] < Low[5] but the rules Open[1] > High[5] and Open[1] < Low[5] can both be true at the same time (therefore the system would show 0 trades as Kantu would exit when it a contradictory long/short signal). In order to avoid this problem you need to ensure that the reverse rule cannot be true at the same time, in this case defining two rules as Open[1] > High[5] and Open[1] > Low[5] would fix the problem.

Remember that SL, TP, hourly filters and daily filters can also be defined for your trading strategy. These values can be defined within the appropriate edit boxes or they can be left blank if they won't be used. If you want to use an AFM-TL you should also make sure that this checkbox is checked within the Simulation options. The image below summarizes some of the functions of the "Run a custom single system" dialog box.

Exporting systems outside Kantu

Notes about back-testing between different platforms

Although simulations between different platforms when using Kantu strategies will look very similar, simulations ran in Kantu will not exactly match the simulations made on other trading platforms due to several key differences in the way in which the simulations are handled. Here are a few reasons why your results will not be exactly the same:

- Kantu uses no swaps and therefore it doesn't have any type of interest (either charged or credited) for any trading strategy. This will cause a difference with platforms that consider swap (either accurately or inaccurately).
- Kantu uses a 100K capital.
- Kantu uses a fixed spread through the whole simulation; a platform that considers explicit Bid/Ask feeds or varies the spread through the simulation in any way will reach different results.

It is also important to remember that Kantu balance graphs are drawn with balance as a function of dates (not trade number) and therefore the graphs will look different than those obtained with MT4. In order to compare the trading results graphically it is necessary to either plot Kantu strategy results as a function of trade number (you can export trades using the trade analysis tab and do this manually) or to plot the MT4 results as a function of the date (Asirikuy users can do this with ASTA).

General notes on exporting

System exporting is done through the "Selection" program menu. In order to export a strategy you must first select it in the same way as if you wanted to plot the strategy's balance graph (click its value on the "No." column in the results grid). Once the system is selected you will be able to export it outside of Kantu using any of the available options or look at the pseudo code by selecting the "Show Pattern Decomposition" option.

Exporting results to MT4

Results can also be exported to an MT4 EA by using the "Export to MQL4" option. This creates an MQL4 file with the target given name. The MQL4 file created has the following available external parameters:

- OPERATONAL_MODE : 0 = normal trading, 1 = only monitor trades for closes, 2 = testing (no user interface is enabled)
- SLIPPAGE : maximum allowed slippage
- INSTANCE_ID : select the "magic number" for the strategy
- DISABLE_COMPOUNDING : set to true to disable compounding within the EA
- ATR_PERIOD : sets the ATR period used for money management (Kantu uses 20)
- RISK : risk as a percentage of the account for each 200% of the ATR
- TAKE_PROFIT : Take profit value as a percentage of the ATR
- STOP_LOSS : Stop Loss value as a percentage of the ATR
- AFMT_TL : Enables AFM_TL if set to true.
- TRADE_COMMENT : Input the custom comment you would like to see on trades

It is also important to note that although the generated MQL4 files are created in good faith, with tested code, there is no guarantee of correct functioning under live trading conditions. It is your sole responsibility to test the code under demo/live trading conditions and to carry out any modifications needed for its reliable execution.