

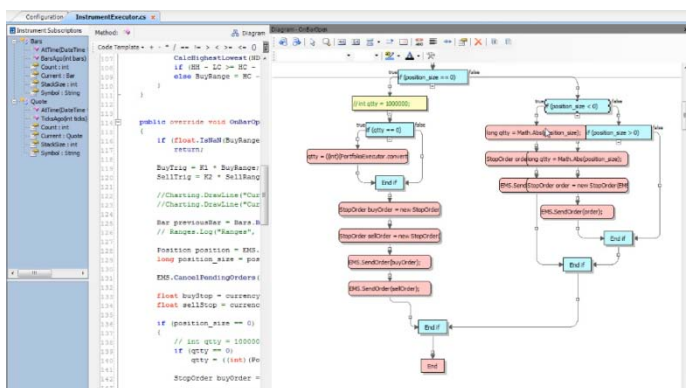
# Product Data Sheet: QuantOffice

## From Strategy Creation Through Production Deployment

QuantOffice is the product for the visual development, debugging and back-testing of integrated Alpha/EMS strategies using C# and Dot Net. It provides a full range of events (eg. OnBarClose , OnBarOpen , OnTick, OnOrderBookChange ) for both portfolio and instrument level granularity and allows combinations of daily, intra-day bar, tick and custom event periodicities to be used in the creation of proprietary order execution algorithms. Once a strategy has been perfected, the strategy (as represented by C# code) is published in QuantServer: UHF Trader for production execution. Time series data for back-testing, simulation and production trading is provided by connectivity to the TimeBase database.

## Visual Strategy Designer

QuantOffice: Studio provides a full C# development environment including two-way integration with Microsoft Visual Studio, Matlab and R. QuantOffice: Studio also provides a rich set of libraries of industry standard technical indicators, statistical and econometric models with which to develop strategies. In addition to creating strategies and models in C#, users can also use the visual process logic builder. Using “drag and drop”, users create a process logic flow diagram; pressing a button then generates the underlying C# code. Conversely, users can generate process logic diagram from the code; that is, they can toggle between the code and graphical representation of the strategy or model.



## Visual Alpha: Wizard-based Model Development

Visual Alpha is special version of QuantOffice which *obviates the need for any C# development*: be that coding or the process logic builder. Users simply use a wizard and select one or more pre-defined indicators, statistical and econometric models from the wizard's library and populate the required parameters. Further, the definition of these strategies can be expressed using a simple lexicon, maintained and manipulated in Excel, and re-run/run in Visual Alpha.

### QuantOffice: Visual Alpha

Model creation and optimization using “drag and drop” of from provided libraries of templates of parameterized indicators and models.

### QuantOffice: Studio

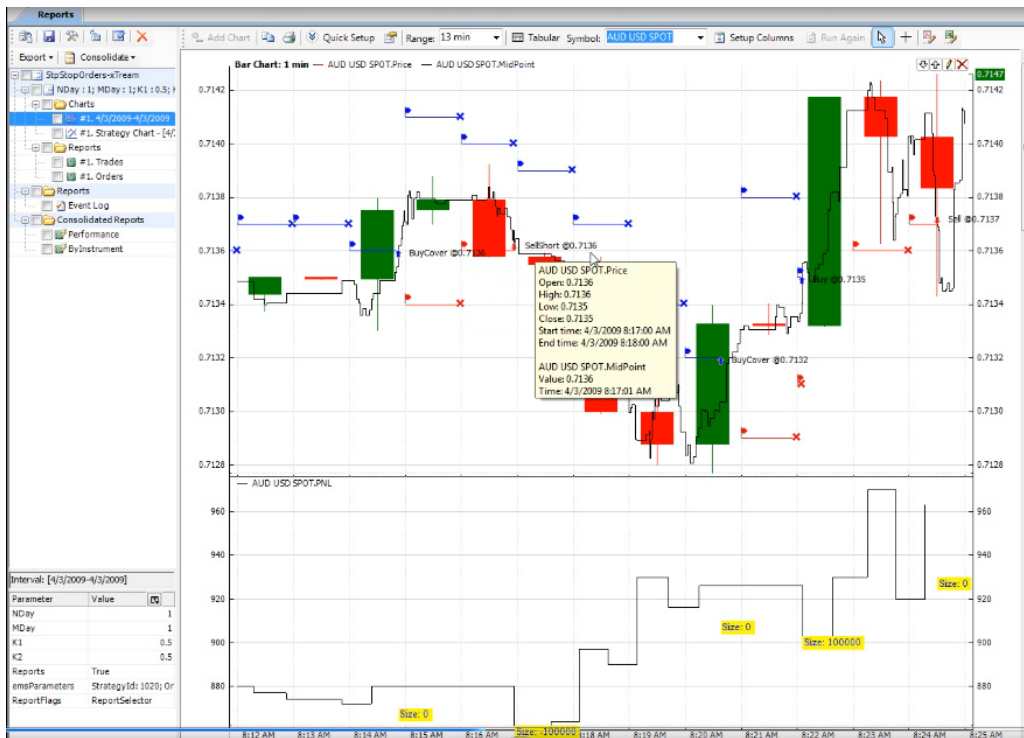
Model creation and optimization using full C# development environment within QuantOffice Studio and/or Microsoft Visual Studio, Matlab or R.

## Complex Event Processing (CEP)

QuantOffice is deployed with TimeBase as its data source. As such, QuantOffice utilizes TimeBase's integrated message bus for CEP. Strategy developers can use provided events (e.g. OnBarClose, OnBarOpen, OnTick) or develop more sophisticated events, e.g. events created as interim output of the strategy itself, or a "meta strategy", that is a model which orchestrates other "sub strategies".

## Rapid Visual Analysis

The output of strategies (indicators, trading signals, orders, executions and P&L) is rapidly displayed graphically, at the instrument and portfolio levels. Microscopic inspection of this output can be performed to see the movement of ticks within bars and the generation of signals, orders and executions at tick periodicities. This allows for rapid evaluation, refinement and re-running of models in an iterative process. The performance of the charting is extraordinary: it takes mere seconds to back-test all instruments in the S&P500 across years' of tick data.



All data can also be shown in tabular form. For example, right mouse-clicking on a single point on a graph, will show all the underlying market data before, on and after that point. All data can be dumped out into Excel, csv and PDF formats.

## Performance

The run-time operation of the models (in back-testing and simulation modes) is immensely fast, as a result of message processing measured in millions of records per second. The output of models operating over hundreds of instruments across years' worth of tick data literally takes seconds. In addition to superior engineering, QuantOffice performance is enhanced by the ability to pre-load events from TimeBase into memory cache.

## Supported Instruments

Equities, options, futures, currencies, pairs, baskets and custom synthetic instruments are supported. Synthetic instruments range from baskets, to the output of strategies themselves.

## Supported Data and Periodicities

Daily, intra-day and tick periodicity data, Level I and Level II (market depth/order book), news and fundamental data are supported. Users can combine different periodicities to construct a strategy. For example, the user may use quarterly fundamental data combined with daily bar data for daily portfolio rebalancing and minute-bar, news or tick data for refinement of execution strategies.

## Bar Creation

In addition to accessing time-based bars created by TimeBase, QuantOffice can be used to create bars using more complex techniques, which are then stored in TimeBase. For example, users can define logic for “by equal volume” or “by equal trade number” bar generation. Such bars are stored in TimeBase in real-time and are accessible to QuantOffice in real-time.

## Optimization

QuantOffice supports optimization of model parameters by brute-force, genetic and dynamic (walk-forward) methods. One of the exciting features of dynamic optimization is the ability to define a “meta strategy”, that is, a strategy which controls, in respect of when they run, other “sub” strategies created in QuantOffice.

## Trading Calendars, Trading Sessions and Exchanges

QuantOffice maintains trading calendars and holidays for all exchanges. Within these constraints, users can define custom trading sessions such as different trading intervals, “no trading” days and continuous 24-hour trading. For synthetic instruments, QuantOffice automatically defines trading session intervals as the intersection of trading sessions for the source exchanges of the synthetic instrument.

## Strategies and Accounts

Strategies can be defined as “sub” strategies of “meta” strategies. Likewise, trading accounts can be grouped into master accounts.

## Reporting Engine

QuantOffice comes with a set of standard reports including trade (order and execution) reports, performance (P&L, drawdown, Sharpe, Sortino etc). There are various definable report criteria such as time interval, strategy, strategy groups as well as the ability to create user-defined reports.

## Production Deployment

Once created and optimized, models which are ready for production deployment are *deployed as is* in QuantServer: UHF Trader. As such, there is no “model deployment risk” which can occur when a model is re-engineered in the production trading environment.

In addition, having deployed the model in production, users can see the actual performance of the model against real-time market data in QuantOffice – the same environment where the model was created and optimized.

## “Warm Up Mode”: Seamless Transition from Simulation to Production

A classic issue when moving to production deployment is that the strategy needs to “know” all its time series indicators based on previous historical data. QuantOffice can run in “warm up mode”, in which state it constantly checks the “strategy time” against market data timestamps. Thus, QuantOffice runs the strategy on historical data until real-time data is reached at which time trading signals seamlessly generate orders for real-time trading.

## Order Execution Management

Particularly in high frequency trading strategies with low margins per trade, the performance of any alpha generation is highly dependent on the success of the translation of trading signals into executed transactions with minimal slippage. In order to enable close coupling between alpha generation and order execution, QuantOffice has order execution capability where users can define their own, or embed, execution algorithms.

## **Trading Simulator**

A key challenge in the successful creation of alpha generation strategies is minimizing the difference in the returns observed during back-testing and those returns actually achieved during live trading. In addition to avoiding “overfitting” the strategy to the training data set, a key component in achieving consistency of returns between back-testing and production is an effective trading simulator. The QuantOffice trading simulator enables precise control of trading assumptions, e.g. specifying the number of ticks which elapse between order creation and execution, percentage order completion.

## **Architecture and Design**

QuantOffice is written in C# and runs on Windows. A rich and documented API allows for direct interaction with the QuantOffice environment. Users can also create their models, or use existing models, written in C++ and use the QuantOffice API to integrate them into the QuantOffice environment, with consequential full access to TimeBase and QuantServer.