

# The Deltix Product Suite: Features and Benefits

## A Product Suite for the full Alpha Generation Life Cycle

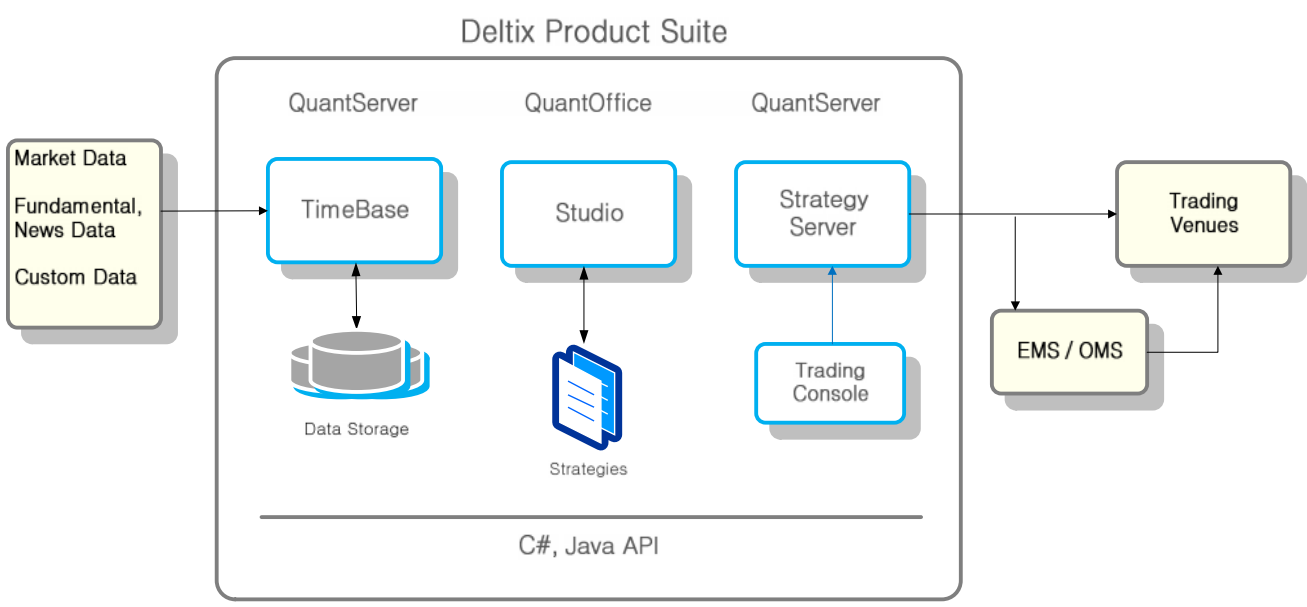
The Deltix Product Suite allows quantitative investors and traders to develop, deploy and manage quantitative trading strategies.



Deltix Product Suite provides end-to-end support of all phases of the alpha discovery process, including data collection and aggregation, model development, back-testing, simulation and deployment to production.

Research:	Data Acquisition and Preparation:	Connect to financial data providers to access historical and real-time data feeds
		Import proprietary financial data
	Alpha Discovery:	Develop models to research market behavior and build proof of concept trading strategies
		Create algorithmic execution strategies
		Back-test and optimize trading and execution models
Production:	Production deployment of models:	Simulate trading on real-time data
		Deploy the trading and execution models as same code
	Monitor and Manage:	Monitor real-time order and executions; P&L and risk
		Perform post-trade portfolio analytics

The diagram below illustrates the Alpha Generation Lifecycle as implemented by the Deltix Product Suite.



The Alpha Generation Lifecycle is often implemented with a combination of software products from different vendors. Whilst this can work, this approach has a number of drawbacks versus the benefits of a single-vendor platform:

Software Optimization:	A single vendor platform is optimized for performance: this does not occur with products from multiple vendors
Inter-operability:	A single vendor platform takes full advantage of all the features in the underlying modules: this is difficult to achieve with a multiple vendor solution
System Implementation:	A single vendor platform requires less time and cost to implement than a multiple vendor solution

## **Rapid Alpha Generation Research**

An easy and intuitive model development framework (*either C# programming and/or* visual “drag and drop”) combined with the integrated time-series database and visual rendering of results make for rapid model development.

## **Integrated Alpha Discovery and Execution Strategy Solution**

The Deltix Product Suite has closely-integrated alpha generation and order execution management for strategies that may include thousands of instruments, seamlessly combining alpha discovery with order execution strategies for superior results.

## **Multiple Instrument Types**

Equities, options, futures, currencies, baskets and custom synthetic instruments are supported.

## **Unparalleled Performance**

Exceptional performance capabilities enable rapid analysis of thousands of instruments for daily, intraday and tick data.

## **Standard Infrastructure**

No specialized hardware is required. Deltix software runs on standard Windows machines, or Linux for some modules. The software is written in C# and Java.

## **Open and Flexible Architecture**

An open architecture environment allows seamless and robust integration with multiple data feeds (e.g. Bloomberg, Thomson/Reuters etc.) and best-in-class software solutions including those for tick processing order execution management and statistical analysis (e.g. Matlab).

Because the Deltix Product Suite is modularized, it is highly flexible. When required, modules of the product suite can be replaced by integration with a client’s own systems or services. For example, while Deltix provides financial database capabilities, QuantOffice can also pull data from client's databases, whether relational or time-series-oriented. This can be controlled at configuration level, making it transparent to the trading algorithm where the data comes from.

## **API**

A rich and documented API allows for direct interaction with TimeBase, QuantOffice and QuantServer. For example, 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.

## Modules for Specific Situations

The Deltix Product Suite comprises modules (all developed in-house by Deltix) which have optimal interaction with each other, but which also can be deployed as modules for specific situations.

<b>QuantServer TimeBase</b>	High (and low) frequency time-series database
<b>QuantOffice Studio</b>	C# model development and optimization and API
<b>QuantOffice DCS Edition</b>	Cloud version of QuantOffice Studio, supplied with access to historical market data (bars)
<b>QuantServer Strategy Server</b>	Order generation and execution engine
<b>QuantServer Trading Console</b>	Monitors for trade execution, P&L and performance

# Product Data Sheet: TimeBase

## Time Series Database

TimeBase is a high performance event-oriented time-series database engine and messaging middleware. TimeBase is designed explicitly for very fast population and retrieval of massive volumes of time-series data and delivering that data for subsequent use by both QuantOffice and third-party software. High volumes data such as fundamental data, news data, daily, bar, tick and Level II (depth of market data) are handled with superior speed. Processing speeds are measured in millions of messages per second on terabytes of data. TimeBase can be populated with historical data and receive real-time data feeds. TimeBase has a sophisticated time-series join engine, capable of efficient on-the-fly merging of multiple data streams with arbitrary temporal characteristics into a unified query response. This capability is central to the TimeBase architecture and is available regardless of the nature of the data source.

## Data Feeds

TimeBase supports digitized news, fundamental data, Level I and Level II data. Many market data vendors and exchanges are supported.

Multiple data feeds can populate TimeBase simultaneously, even for the same instrument.

## Instruments

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

## Data Periodicities and Regularity

Any periodicity of data can be handled: down to 1 millisecond intervals such as tick and Level II data. Data can be both *regular* and *irregular*. For example, irregular tick data can be maintained and converted into bar data (regular).

## Bar Creation

When recording tick or intra-day data, TimeBase can create and maintain time-based bars automatically. In QuantOffice, users can 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.

## Data Population

TimeBase supports historical population, backfilling and real-time data feeds. Deployed with QuantOffice, TimeBase can be used for *both back-testing and managing strategies in production*.

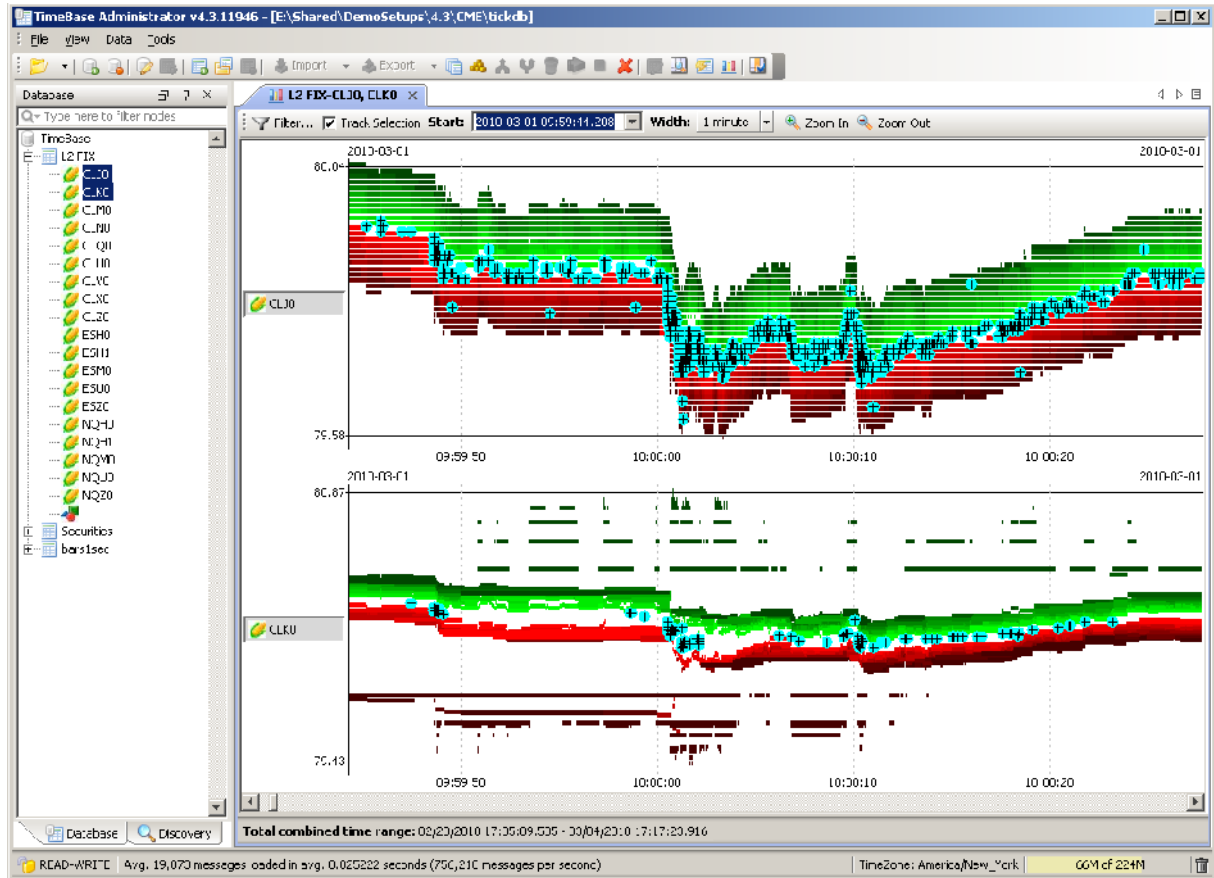
## User Interaction

TimeBase has a Graphical User Interface (GUI), called TimeBase Administrator, which provides system and data configuration and graphing capabilities which are capable of displaying years' worth of tick data across thousands of instruments in seconds. With the GUI, the user can zoom into the graphs to inspect data at the most granular level. All graphs and the raw data can be downloaded.

TimeBase Administrator - Data View

Time	deltix.qsrv.hf.pub.TradeMessage		deltix.qsrv.hf.pub.BestBidOfferMessage			
	Price	Size	Bid Price	Bid Size	Offer Price	Offer Size
01/02/2009 09:30:31.000			0.000000	0.000000	84.660000	1.000000
01/02/2009 09:30:31.000			84.610000	2.000000	0.000000	0.000000
01/02/2009 09:30:31.000	84.630000	74,300.000000				
01/02/2009 09:30:31.000			84.470000	4.000000	0.000000	0.000000
01/02/2009 09:30:31.000			84.400000	50.000000	0.000000	0.000000
01/02/2009 09:30:32.000			0.000000	0.000000	84.660000	2.000000
01/02/2009 09:30:32.000			84.450000	1.000000	0.000000	0.000000
01/02/2009 09:30:32.000			84.450000	2.000000	0.000000	0.000000
01/02/2009 09:30:32.000			84.400000	50.000000	0.000000	0.000000
01/02/2009 09:30:32.000			84.480000	20.000000	0.000000	0.000000
01/02/2009 09:30:32.000			84.410000	1.000000	0.000000	0.000000
01/02/2009 09:30:32.000			84.410000	3.000000	0.000000	0.000000
01/02/2009 09:30:32.000			84.520000	1.000000	0.000000	0.000000
01/02/2009 09:30:32.000			84.540000	1.000000	0.000000	0.000000
01/02/2009 09:30:34.000	84.650000	200.000000				
01/02/2009 09:30:34.000			84.650000	2.000000	0.000000	0.000000
01/02/2009 09:30:34.000	84.660000	100.000000				
01/02/2009 09:30:34.000			84.540000	1.000000	0.000000	0.000000
01/02/2009 09:30:41.000	84.630000	100.000000				
01/02/2009 09:30:41.000			84.630000	1.000000	0.000000	0.000000
01/02/2009 09:30:41.000			84.540000	1.000000	0.000000	0.000000
01/02/2009 09:30:42.000			0.000000	0.000000	84.660000	1.000000
01/02/2009 09:30:42.000	84.660000	100.000000				
01/02/2009 09:30:42.000			84.630000	1.000000	0.000000	0.000000
01/02/2009 09:30:42.000			0.000000	0.000000	84.660000	2.000000
01/02/2009 09:30:42.000			0.000000	0.000000	84.709900	2.000000
01/02/2009 09:30:43.000	84.709900	100.000000				
01/02/2009 09:30:43.000			0.000000	0.000000	84.709900	1.000000
01/02/2009 09:30:45.000	84.709900	100.000000				
01/02/2009 09:30:47.000			0.000000	0.000000	84.709900	2.000000
01/02/2009 09:30:58.000			84.560000	1.000000	0.000000	0.000000
01/02/2009 09:30:58.000	84.630000	100.000000				
01/02/2009 09:30:58.000			0.000000	0.000000	84.650000	16.000000
01/02/2009 09:30:58.000			84.540000	1.000000	0.000000	0.000000
01/02/2009 09:30:58.000			0.000000	0.000000	84.650000	17.000000
01/02/2009 09:30:58.000			0.000000	0.000000	84.650000	18.000000
01/02/2009 09:30:59.000	84.630000	100.000000				

## TimeBase Administrator – Graph View



## Deltix Product Suite

Together with the QuantOffice products, TimeBase is a core component of the Deltix Product Suite that addresses all stages of the Alpha Generation Process. As such, TimeBase is either deployed as part of a QuantServer and QuantOffice deployment, or stand-alone, integrated with a client's in-house or third-party applications.

## Architecture and Design

TimeBase provides diverse ways of specifying data types, as well as hints for encoding values for transmission and persistence. One benefit of such diversity is a more native representation of user's data model. For example, TimeBase natively supports enumerations. As an example, consider a data item whose value is 'SPOT', 'FORWARD', 'SWAP' or 'N/A', in TimeBase, one can define an enumeration, and everyone will always know what the allowed values are. Additionally, this data item can map to native Java or C# enumerations for programming clarity.

The use of rich data types enhances system performance. This is achieved by TimeBase knowing the detailed data type of every value, and the user's ability to define at a very fine-grained level how the data should be encoded. TimeBase can then allocate less resource (memory, network bandwidth, disk space) and process data quicker. In the above example of an enumeration, each enumerated data

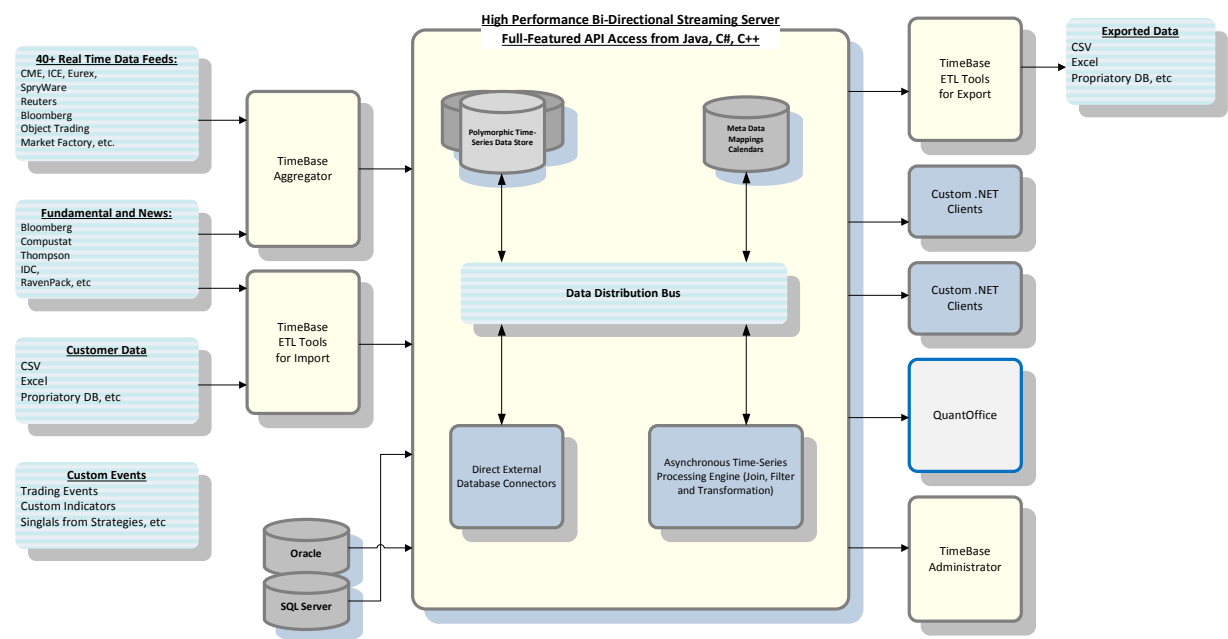
item only requires 2 bits to store its value (there are only 4 possible values), even if it appears to the user as a text constant.

TimeBase supports special compression for small alphanumeric codes which allows the data item to appear as text, while it is internally stored in extremely compressed form. Examples of alphanumeric codes include exchange codes, currency codes, liquidity provider codes, or any other codes the user desires to define.

TimeBase supports so-called relative encoding. In a price bar, it makes sense to store close as a number, and open, high, low as a delta from close. This saves space and increases speed. In TimeBase the relative encoding is a generic part of the type system and can be used with any user-defined data structure, not only price bars.

TimeBase has a built-in message broker. All interactions with the database, either via Deltix products, user or third-party applications, are handled by this message broker.

## TimeBase Logical Design



## Hardware

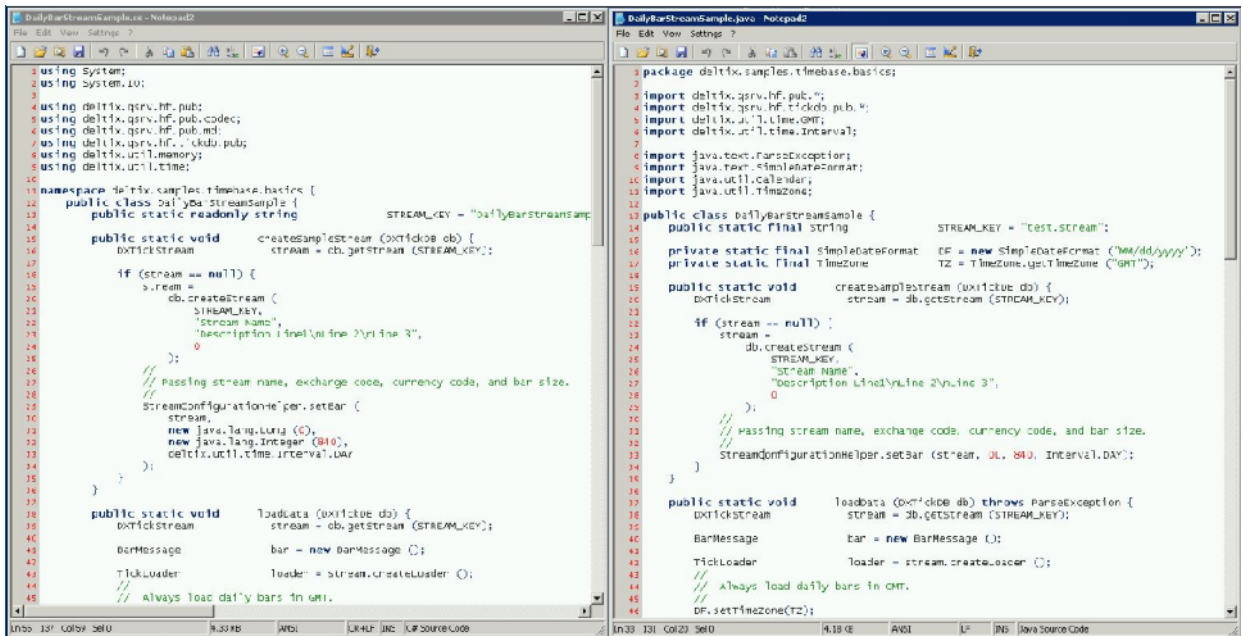
It is a common misconception that high performance requires either accelerated or very high end (that is, expensive) hardware. It does not. But, high performance does require superior software engineering: a skill often forgotten in a world of ever more powerful computers. TimeBase is written in Java, and so runs on Windows and Linux platforms.



## API

TimeBase has a rich API, against which C#, C++ and Java code can be written. Examples are shown below:

### Interfacing with the TimeBase<sup>™</sup> API: C# and Java code examples



The image displays two side-by-side screenshots of code editors, illustrating C# and Java code examples for interfacing with the TimeBase API.

**Left Editor (C#):** The code is in a file named `DailyBarStreamSample.cs`. It uses the `System` and `System.IO` namespaces. It defines a `namespace deltax.samples.timebase.basics` containing a `public class DailyBarStreamSample`. The class has a `public static readonly string STREAM_KEY = "dailybarstreamsample";`. It includes a `createSampleStream(Dxtickdb db)` method that creates a `DxtickStream` object, sets its `StreamName`, `ExchangeCode`, `CurrencyCode`, and `BarSize`. It also includes a `loadData(Dxtickdb db)` method that creates a `BarMessage` object, sets its `StreamName`, `ExchangeCode`, `CurrencyCode`, and `BarSize`, and then loads the data into the `Stream` object.

**Right Editor (Java):** The code is in a file named `DailyBarStreamSample.java`. It uses the `deltax.samples.timebase.basics` package. It defines a `public class DailyBarStreamSample`. The class has a `public static final String STREAM_KEY = "test.stream";`. It includes a `createSampleStream(Dxtickdb db)` method that creates a `DxtickStream` object, sets its `StreamName`, `ExchangeCode`, `CurrencyCode`, and `BarSize`. It also includes a `loadData(Dxtickdb db) throws ParseException` method that creates a `BarMessage` object, sets its `StreamName`, `ExchangeCode`, `CurrencyCode`, and `BarSize`, and then loads the data into the `Stream` object.

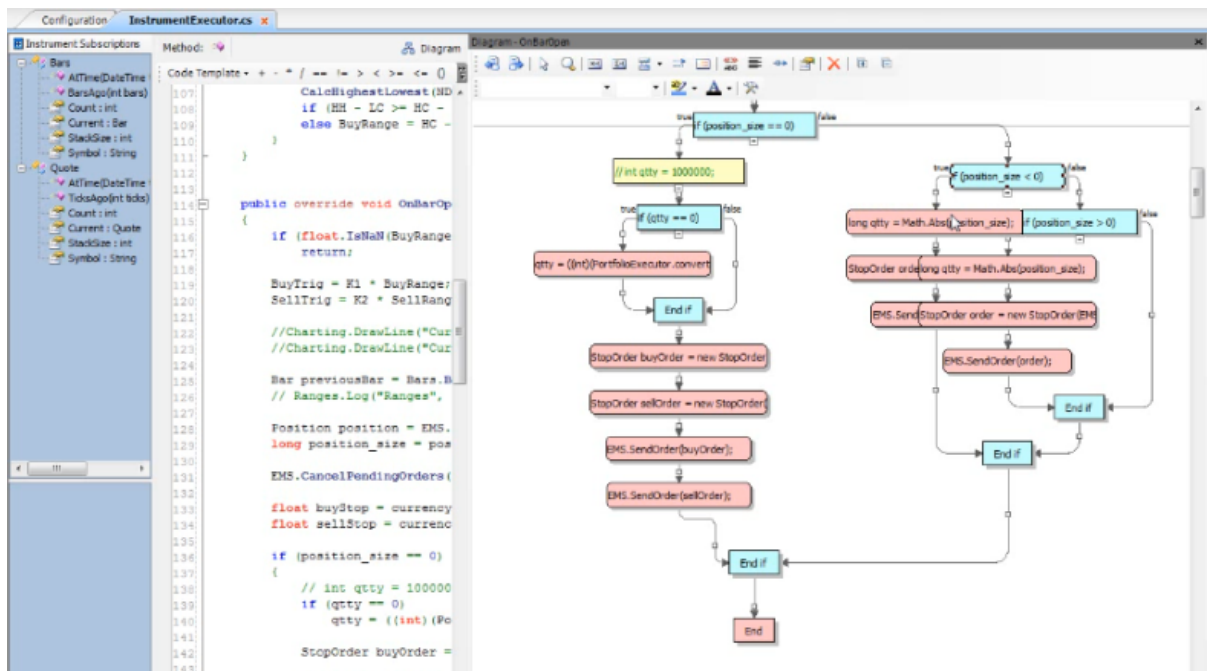
# Product Data Sheet: QuantOffice

## 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 Strategy Trading Server 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.

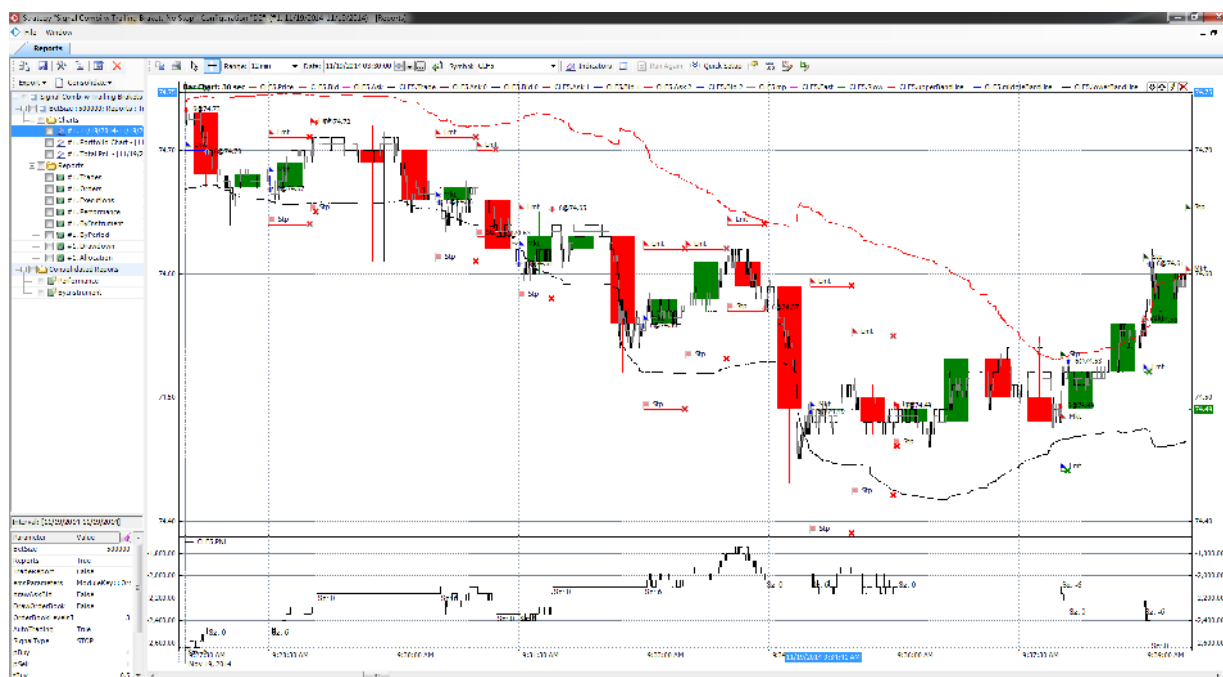


## MS Visual Studio, Matlab, Python and R Compatibility: Round-trip Engineering Complex Event Processing (CEP)

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 Strategy Trading Server. 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.

# Product Data Sheet: QuantServer

## Production Deployment of Strategy

QuantServer consists of **Strategy Server** (formerly known as “UHF”) and **Trading Console**: the two modules required for production deployment of trading strategies created in QuantOffice. Once created, a strategy is “published” in StrategyServer (STS). STS converts trading signals into FIX orders and executes them according to the execution strategy defined in QuantOffice, potentially via multiple brokers. The progress of the trading (that it is: status of orders, execution, P&L and performance) are monitored in Trading Console.

StrategyServer is designed for very high performance, processing hundreds of thousands of messages per second, arising from hundreds of instruments and hundreds of strategies.

## Strategy Deployment: Live vs. Simulated Trading vs. Back-Testing Transparency

The same trading strategy created in QuantOffice is deployed in production for live trading “as is”. The switch between the trading simulators, paper trading and live trading is transparent to the trading strategy. With a single click of a button, a user-defined strategy is uploaded to the server, instantiated and hooked up to the data feeds and trading services. There are graphical tools for deploying algorithms to remote data centers.

## Market Data Transparency

StrategyServer is provided with real-time (and historical) data from TimeBase. As such, switching market data vendors is transparent for trading strategies. The strategy determines which data is required; this data is provided from TimeBase.

## API

Trading models have access to QuantServer’s API for order creation and management. The API is vendor-independent and provides complete access to the capabilities supported by the FIX protocol, as well as the ability to utilize individual vendor extensions, if so desired.

## Risk Management

A dedicated Risk Monitor screen can be used to set trading limits such order size, position size, number of positions, order submission rate etc. Limits can be set at different levels, e.g. broker, trader.

## Time Management

The strategy can set a timer to run, for example, 30 minutes before the market opens, or every 30 seconds. StrategyServer will “wake up” the strategy to perform required work.

## Infrastructure

StrategyServer is written in Java, and so runs on Windows and Linux platforms. A standard Windows server or workstation can process all of North American Level I equity and option data in real-time.

## Trading Console

Trading Console is a set of real-time monitors of order and execution status, P&L and performance, risk and portfolio analytics. The trade blotter monitor, allows for “grey box trading”, that is, users may interact with the model by cancelling some or all of the orders created by the model.

