

# ***S t o c k F u s i o n***

*Industrial strength solutions for Financial Forecasting*

# **ForeStock**

---

**WINNING TRADING STRATEGIES FOR  
EQUIS METASTOCK®, TRADE STATION®,  
NINJA TRADER, MULTICHARTS®  
AND MICROSOFT EXCEL®**

*by Boris G. Zinchenko, Ph.D.*

October 2010

## Contents

---

Contents .....	2
Introduction .....	3
Installation.....	4
System requirements .....	4
Institutional architecture.....	5
Algorithms.....	7
Trading platforms.....	10
TradeStation.....	10
Metastock.....	12
Ninja Trader .....	14
MultiCharts .....	15
Excel .....	15
Algorithm settings and limits.....	18
References .....	21

## Introduction

---

ForeStock is the original name for the completely new family of market forecasting technologies. ForeStock stands for "**Forecast Stocks**", "**Foresee Stock Market Changes**", "**Forecast Stochastic Price Charts**" or whatever else sensible abbreviations, which you may imagine. It has nothing to do with the firearms, even though is designed to be as precise as a sniper rifle in targeting market trends!

ForeStock is a predictor and indicator module that plugs into TradeStation, MetaStock, Ninja Trader and Excel. ForeStock a number of highly adaptive cutting edge market predictors. In addition to predictors, ForeStock contains a large number of specialized indicators suited for various market conditions.

ForeStock is a completely new way of looking at the markets. ForeStock uses highly advanced mathematical principles to identify patterns in price, bid/ask, time and volume that otherwise appears completely random. Using advances in fractal mathematics, ForeStock can identify trends that are just beginning to form.

ForeStock is not a system, it is an easy to use module that allows anyone familiar with the compatible platforms to build personal and "private" trading systems. Use the predictors or indicators any way that you see fit. Possibilities include best of breed, predictor reversion, spread models, predictor pricing models, etc.

Professional traders and even part time investors will find ForeStock a powerful tool to add to their trading arsenal. Experienced traders and money managers will find ForeStock incredibly easy to tailor for specific investment portfolios. A complete list of the predictors and analytical tools is described in this guide book.

## Installation

---

To install ForeStock please follow these steps:

1. Download ForeStock directly from ForeStock web site
2. Run automated installer and follow instructions on screen
3. Request user name and password to access all features
4. Install any supported trading terminals and follow instructions for your platform in sections below.

Installation on Windows Vista requires administrative user privileges. Installed product can run then under regular user credentials.

### **System requirements**

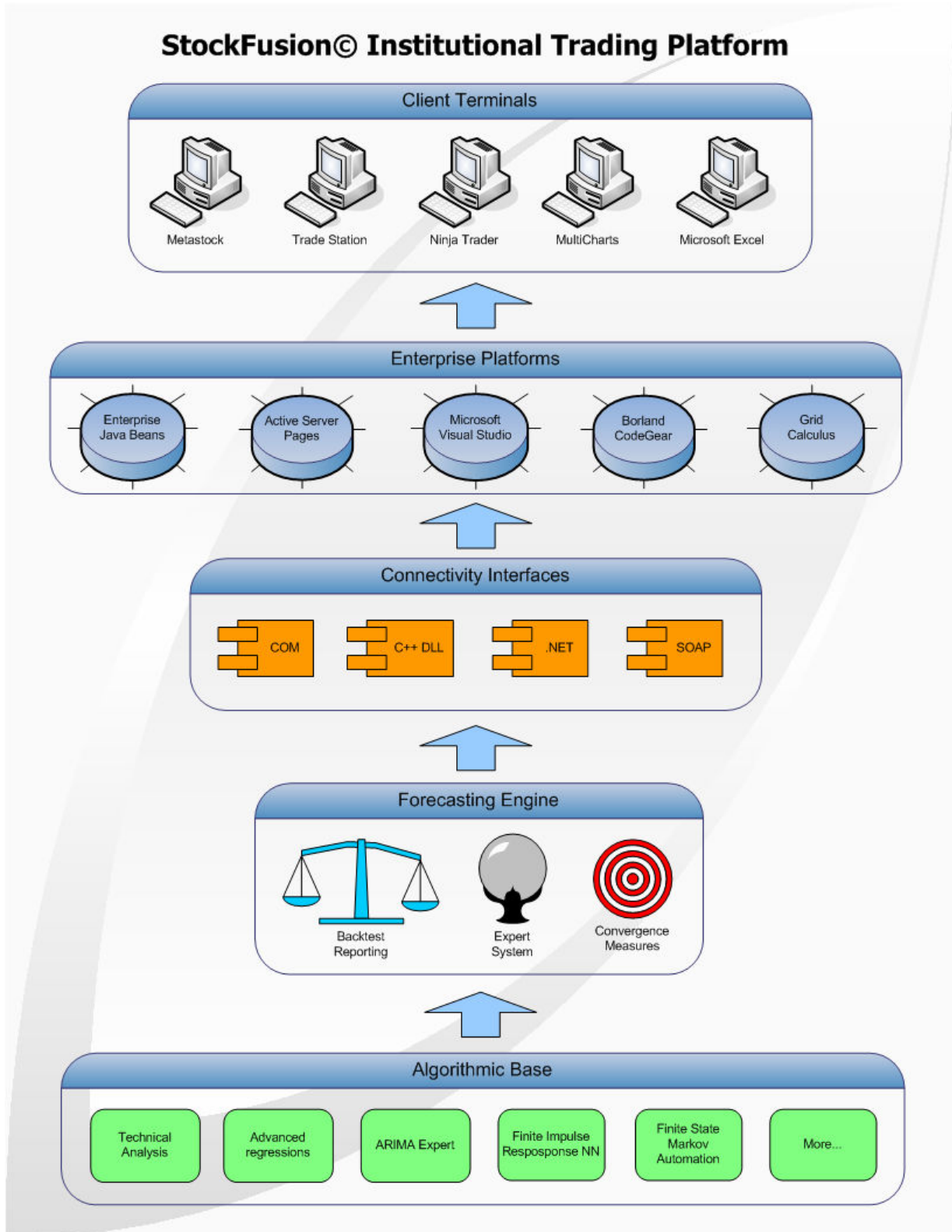
Microsoft Windows 2000, XP, 2003 Server, Vista, 2008 Server, Windows 7

Intel x86 compatible computer with Pentium 4 processor 1 GHz or better, RAM 256 MB, 10 MB disk space.

While system requirements for running application are minimal, its real resource consumption will totally depend on resources required by trading terminal and the frequency of the data frame. Trading on short intraday signals will require top performance computers available on the market. Even in this case we cannot guarantee that system resources will be sufficient for stable and timely delivery of forecasts and trading signals. We do not bear any responsibility for any delays realted with insufficient performance of client hardware.

## Institutional architecture

ForeStock offers powerful system integration abilities invaluable for institutional investors and trading houses. It seamlessly and easily integrates into virtually any proprietary trading software and existing trading solutions. Diagram below shows the rich integration abilities available with ForeStock platform.



# **ForeStock**

ForeStock offers following APIs:

1. C++ export library (fastest)
2. COM Server
3. .NET 2.0 API
4. SOAP API for remote forecasting service deployment

With this set of industry standard interfaces, blazing speed of native C++ compilation and ready integration templates for all leading enterprise environments, ForeStock provides you with the instant ability to reinforce your existing proprietary trading solutions with its innovative market forecasting technologies.

## Algorithms

Below we provide the full table of algorithms available in ForeStock. Algorithms are grouped according to their packages. Packages are separate modules in common algorithmic space and are licensed individually. You can watch names of packages in License Manager. Licensing any package implies all algorithms contained inside it.

Forecasting algorithm	Description	Type
<b>ARIMA Expert</b>		
ARIMA with expert model fit	Seasonal Auto-Regressive Integrated Moving Average forecasting model with automatic expert inference on all model parameters.	Predictor
<b>Finite State Markov Automation</b>		
Finite State Markov Automation	We dynamically construct Markov models that describe the characteristics of Market data flow. Such models are used to predict future market states.	Predictor
<b>Finite Impulse Response NN</b>		
Finite impulse response neural network	The finite impulse response neural network is a neural network, where scalar weights are replaced with moving average filters. These filters compute a weighted average of past values presented to the network, as opposed to the feed-forward network, which only computes a weighted "average" of the current value. These networks are trained using a variation on the backpropagation algorithm.	Predictor
<b>Advanced Regressions</b>		
Forecast with average value	Classical moving average with period 20	Predictor
Linear regression	Linear regression line <b><math>y = at + b</math></b> calculated over 20 last points	Predictor
Exponential Fit	Exponential regression curve <b><math>y = e^{at + b}</math></b> calculated over 20 last points	Predictor
Logarithmic Fit	Logarithmic regression <b><math>y = \log(at + b)</math></b> calculated over 20 last points	Predictor
Logistic Fit	Logistic regression <b><math>y = c / [1 + e^{-(at + b)}]</math></b> calculated over 20 last points	Predictor
Square Fit	Parabolic regression <b><math>y = (at + b)^2</math></b> calculated over 20 last points	Predictor
Square Root Fit	Square root regression <b><math>y = (at + b)^{1/2}</math></b> calculated over 20 last points	Predictor
History Prophet	Emulates "ideal" predictor. Forecast is set to real next observed value, which ensures 100% forecasting accuracy	Predictor

	on historical data. It is very useful to calibrate performance of trading strategies in "ideal" conditions. In no case, it should be used as predictor in real trading.	
Naive Predictor	Forecast with the previous price. Dummy forecast to evaluate performance of other algorithms.	Predictor
<b>Technical Analysis</b>		
1 day ROC of 3 Smooth EMA		Indicator
Absolute Price Oscillator		Indicator
Aroon Down		Indicator
Aroon Oscillator		Indicator
Aroon Up		Indicator
Average Price		Indicator
Average True Range		Indicator
Avg Direct Move Idx Rating		Indicator
Avg Directional Move Idx		Indicator
Chaikin AD Line		Indicator
Chaikin AD Oscillator		Indicator
Commodity Channel Index		Indicator
Directional Movement Index		Indicator
Double Exponential MA		Indicator
Exponential Moving Average		Indicator
FXS Adaptive Moving Average		Indicator
Hilbert Dominant Cycle Period		Indicator
Hilbert Dominant Cycle Phase		Indicator
Hilbert Phasor Components		Indicator
Hilbert SineWave		Indicator
Hilbert Transform Trendline		Indicator
Hilbert Trend vs Cycle Mode		Indicator
Kaufman Adaptive MA		Indicator
Linear Regr Angle		Indicator
Linear Regr Intercept		Indicator
Linear Regr Slope		Indicator
Linear Regression		Indicator
MESA Adaptive Moving Average		Indicator
Median Price		Indicator
MidPoint over period		Indicator
Midpoint Price over period		Indicator
Minus Directional Indicator		Indicator
Minus Directional Movement		Indicator
Momentum		Indicator



Money Flow Index	Indicator
On Balance Volume	Indicator
Parabolic SAR	Indicator
Percentage Price Oscillator	Indicator
Plus Directional Indicator	Indicator
Plus Directional Movement	Indicator
Rate of change	Indicator
Rate of change Percentage	Indicator
Rate of change ratio	Indicator
Relative Strength Index	Indicator
Simple Moving Average	Indicator
Time Series Forecast	Indicator
Triangular Moving Average	Indicator
Triple EMA	Indicator
Triple EMA T3	Indicator
True Range	Indicator
Typical Price	Indicator
Weighted Close Price	Indicator
Weighted Moving Average	Indicator
Williams Percent R	Indicator

## Trading platforms

---

ForeStock runs on the multitude of trading platforms. Installation instructions are specific for each platform. Please find your platform in the list below and follow instructions for it.

### TradeStation

To deploy ForeStock on TradeStation, please follow these steps:

1. Install TradeStation 2000i or above
2. Install ForeStock
3. Enter license keys for all components
4. Run ForeStock setup for Trade Station

**Programs > StockFusion > StockFusion for Trade Station**

5. There will appear import wizard screen. Select all objects for import and confirm.
6. Watch imported functions in Easy Language editor. Use provided functions as templates for your own functions and strategies.

Below there is full Easy Language code of Aura Forecast Engine call.

```
{*****
  Description: Aura Forecast Engine Extended
  Provided By: Boris Zinchenko (c) Copyright 2008
*****}

DefineDLLFunc: "EEOmegaX.dll", float, "AURA_ENGINE_EX", LPSTR, DWORD, DWORD, LPLONG, LPINT,
LPLONG, LPLONG, LPLONG, LPLONG, LPLONG, LPLONG, LPSTR, LPSTR, DWORD;

{ Inputs }

Inputs:
AlgorithmName(String),
InputLength(Numeric),
SeriesNames(String),
Parameters(String),
ForecastLength(Numeric),
Forecast(NumericRef);

{ Inner variables }
Variables: Dummy(0), Counter(0);

{ Reserve arrays for data }
Dummy = Date[InputLength];
Dummy = Time[InputLength];
Dummy = Open[InputLength];
Dummy = High[InputLength];
Dummy = Low[InputLength];
Dummy = Close[InputLength];
Dummy = Volume[InputLength];
Dummy = OpenInt[InputLength];

{ Call solver }
Forecast = AURA_ENGINE_EX((LPSTR)AlgorithmName, (DWORD)MaxBarsBack, (DWORD)PriceScale,
(LPLONG)&Date, (LPINT)&Time, (LPLONG)&Open, (LPLONG)&High,
(LPLONG)&Low, (LPLONG)&Close, (LPLONG)&Volume, (LPLONG)&OpenInt,
(LPSTR)SeriesNames, (LPSTR)Parameters, (DWORD)ForecastLength);

AuraEngineExt = Forecast;
```

Short description of input parameters:

Parameter	Description
-----------	-------------

# ForeStock

AlgorithmName	Name of indicator. It must exactly coincide with the names of algorithms given in algorithm table in this manual. Any misprint, wrong case or white space will result in error.
InputLength	Desirable input length of symbol history for calculation. Please carefully observe minimum input length limitations in the last section of this manual. If input is too short, no calculation will take place and indicator will stay void. Typically, minimal limitations are just barely enough to run algorithm at all. User must expect that good results will require much longer series. It is advised to have at least 500 historical points provided. Best results are expected with several thousands historical points. Note however that the increase of history dramatically increases required computational resources.
SeriesNames	Comma delimited list of input series for calculation. Allowed names include <ul style="list-style-type: none"><li>• Open</li><li>• High</li><li>• Low</li><li>• Close</li><li>• Volume</li><li>• OI</li></ul> For example: <b>SeriesNames("Open,High,Low,Close,Volume,OI")</b> . Names are case sensitive. No whitespaces allowed. User can rearrange names to combine different input sequences to the algorithm. There is no sense to pass several series into algorithms, which are univariate. Please consult algorithm table in the last chapter on allowed number of inputs for each algorithm.
Parameters	Reserved for future usage and complex use cases. Please ignore in current release.
ForecastLength	Number of forward steps, for which forecast is calculated and returned as function output. Please consult algorithm table in the last chapter on supported forecast lengths for each specific algorithm. Indicators support only zero forecast length. Predictors support one or more forward steps.

This is very simple example of calling engine in trading strategy.

```
{*****
  Description: StockFusion Universal Signal
  Provided By: Boris Zinchenko (c) Copyright 2008
  *****}

Inputs: AlgorithmName("Linear Regression"), InputLength(500), SeriesNames("Close"),
Parameters(""), ForecastLength(1);

Variables: Forecast(0);

Value1 = QB_AuraEngineExt(AlgorithmName, InputLength, SeriesNames, Parameters, ForecastLength,
Forecast);

If (Forecast < Close) AND (Close > Close[1]) Then Sell This Bar at Close;
If (Forecast > Close) AND (Close < Close[1]) Then Buy This Bar at Close;
```

Of course, real strategies are typically more complex and realistic .



## Metastock

To deploy on Metastock, please follow these steps:

1. Install Metastock 7 or above
2. Install ForeStock
3. Enter license keys for all components
4. Run ForeStock setup for Metastock

**Programs > StockFusion > StockFusion for Metastock**

After import succeeds, you will watch ForeStock indicators and experts inside Metastock. They all begin with word: "ForeStock - ". There are offered distinct indicators for all algorithms we have and only one truly schematic expert for illustration. All code is open, so user is expected to develop own extensions based on these patterns.

To use ForeStock in Metastock formulas you must call external functions in StckFusion extension DLL. Simple example of such external call:

```
ExtFml ( "EEMetaSt.AuraEngine", ARIMA )
```

To call each algorithm, you must enter its name in formula exactly as given on the table below.

## Predictors

Algorithm	Metastock
ARIMA	ARIMA
Exponential Fit	EXPONENTIALFIT
Finite Impulse Response NN	FINITEIMPULSERESPONSENN
Finite State Markov Automaton	FINITESTATEMARKOVAUTOMATON
Forecast with average value	FORECASTWITHAVERAGEVALUE
History Prophet	HISTORYPROPHET
Linear Regression	LINEARREGRESSION

Logarithmic Fit	LOGARITHMICFIT
Naive Predictor	NAIVEPREDICTOR
Square Fit	SQUAREFIT
Square Root Fit	SQUAREROOTFIT
Stepwise Best Regression MVAR	STEPWISEBESTREGRESSIONMVAR

## Indicators

Algorithm	Metastock
1 day ROC of 3 Smooth EMA	1DAYROCOF3SMOOTHEMA
Absolute Price Oscillator	ABSOLUTEPRICEOSCILLATOR
Aroon Down	AROONDOWN
Aroon Oscillator	AROONOSCILLATOR
Aroon Up	AROONUP
Average Price	AVERAGEPRICE
Average True Range	AVERAGETRUE RANGE
Avg Direct Move Idx Rating	AVGDIRECTMOVEIDX RATING
Avg Directional Move Idx	AVGDIRECTIONALMOVEIDX
Chaikin AD Line	CHAIKINADLINE
Chaikin AD Oscillator	CHAIKINADOSCILLATOR
Commodity Channel Index	COMMODITYCHANNELINDEX
Directional Movement Index	DIRECTIONALMOVEMENTINDEX
Double Exponential MA	DOUBLEEXPONENTIALMA
Exponential Moving Average	EXPONENTIALMOVINGAVERAGE
FXS Adaptive Moving Average	FXSADAPTIVEMOVINGAVERAGE
Hilbert Dominant Cycle Period	HILBERTDOMINANTCYCLEPERIOD
Hilbert Dominant Cycle Phase	HILBERTDOMINANTCYCLEPHASE
Hilbert Phasor Components	HILBERTPHASORCOMPONENTS
Hilbert SineWave	HILBERTSINEWAVE
Hilbert Transform Trendline	HILBERTTRANSFORMTRENDLINE
Hilbert Trend vs Cycle Mode	HILBERTTRENDVSCYCLEMODE
Kaufman Adaptive MA	KAUFMANADAPTIVEMA
Linear Regr Angle	LINEARREGRANGLE
Linear Regr Intercept	LINEARREGRINTERCEPT
Linear Regr Slope	LINEARREGRSLOPE
Linear Regression	LINEARREGRESSION
MESA Adaptive Moving Average	MESAADAPTIVEMOVINGAVERAGE
Median Price	MEDIANPRICE
MidPoint over period	MIDPOINTOVERPERIOD
Midpoint Price over period	MIDPOINTPRICEOVERPERIOD
Minus Directional Indicator	MINUSDIRECTIONALINDICATOR

Minus Directional Movement	MINUSDIRECTIONALMOVEMENT
Momentum	MOMENTUM
Money Flow Index	MONEYFLOWINDEX
On Balance Volume	ONBALANCEVOLUME
Parabolic SAR	PARABOLICSAR
Percentage Price Oscillator	PERCENTAGEPRICEOSCILLATOR
Plus Directional Indicator	PLUSDIRECTIONALINDICATOR
Plus Directional Movement	PLUSDIRECTIONALMOVEMENT
Rate of change	RATEOFCHANGE
Rate of change Percentage	RATEOFCHANGEPERCENTAGE
Rate of change ratio	RATEOFCHANGERATIO
Relative Strength Index	RELATIVESTRENGTHINDEX
Simple Moving Average	SIMPLEMOVINGAVERAGE
Time Series Forecast	TIMESERIESFORECAST
Triangular Moving Average	TRIANGULARMOVINGAVERAGE
Triple EMA	TRIPLEEMA
Triple EMA T3	TRIPLEEMAT3
True Range	TRUERANGE
Typical Price	TYPICALPRICE
Weighted Close Price	WEIGHTEDCLOSEPRICE
Weighted Moving Average	WEIGHTEDMOVINGAVERAGE
Williams Percent R	WILLIAMSPERCENTR

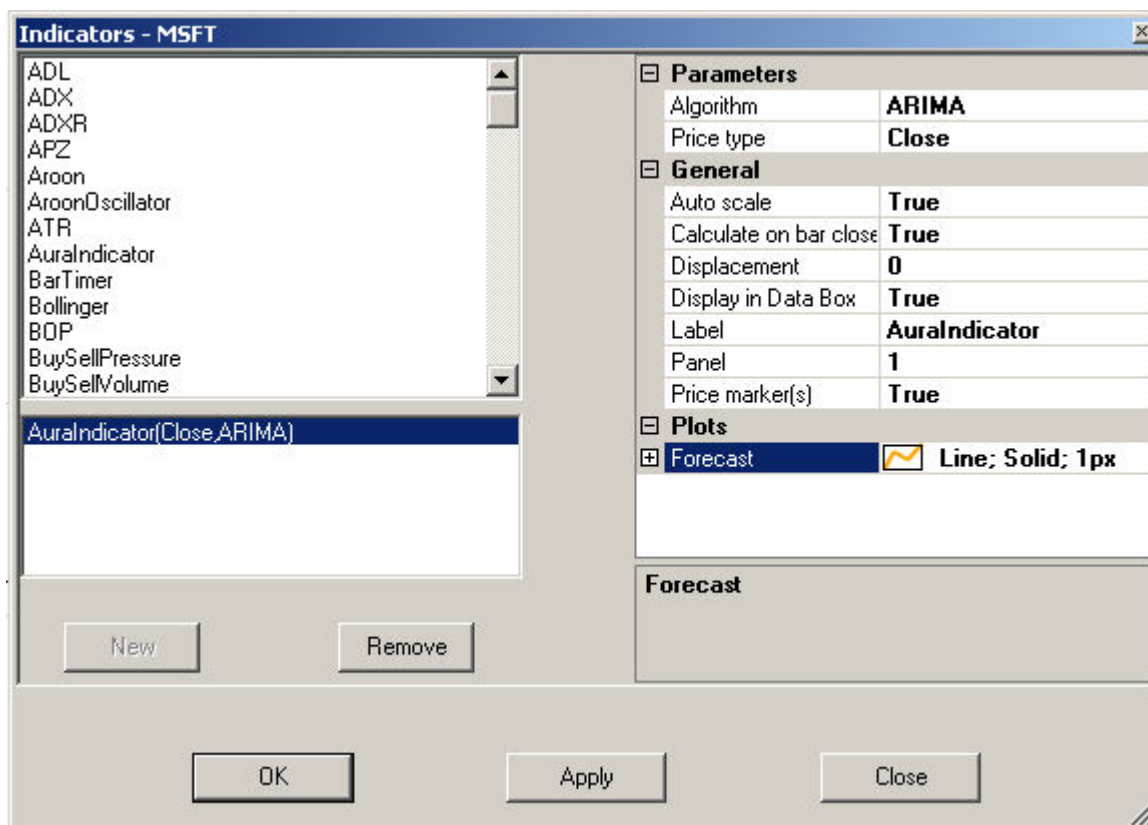
In addition to standard function **AuraEngine**, we offer the extended function **AuraEngineEx**. This function is full analog of **AuraEngine** except it makes real backtest, that is, it recalculates each model on each step fully excluding possibility of forward looking effect in the backtest. Due to this difference, **AuraEngineEx** is longer in calculation than **AuraEngine** exactly number of times as the number of points in series. For example, if the series has 1000 points, then **AuraEngineEx** will take 1000 times more to calculate. Unless you have very powerful computer or do need very scrupulous backtest, we always advise using **AuraEngine** with nearly equal results.

## ***Ninja Trader***

To install on Ninja Trader, please follow these steps:

1. Install Ninja Trader
2. Install ForeStock
3. Enter license keys for all components
4. Open Ninja Trader
5. Use menu: **File > Utilities > Import NinjaScript**
6. Select "EENinja.zip" in setup directory and confirm import
7. Ignore warnings on external assemblies. They are all preconfigured by our installer

Open chart and add indicator to it. Type algorithm name exactly as specified an the list of available algorithms



## MultiCharts

To deploy ForeStock on MultiCharts, please follow these steps:

1. Install MultiCharts.
2. Install ForeStock.
3. Enter license keys for all components.
4. Open the MultiCharts PowerLanguage Editor.
5. Click on the File drop down menu.
6. Click on Import.
7. Navigate to installation directory and locate there file "StockFusion.eld"
8. Highlight and Open it.
9. Import studies will appear with the Function and Signal listed.
10. Click OK to confirm import.
11. Click the Compile drop down menu.
12. Click on All Uncompiled.
13. Open a chart and insert the Signal.
14. To change predictors, rename in the Inputs section of Format Signal.

## Excel

To deploy ForeStock on Microsoft Excel, please follow these steps:

1. Install Microsoft Office 98 or above
2. Install ForeStock

3. Enter license keys for all components
4. Run Excel demo example

## Programs > StockFusion > StockFusion for Excel

This example contains simple VBA code to call algorithms. It is intended rather as illustration. It is expected that users will utilize this example in building real world trading systems on their own.

```
Const colDate = 1      ' dates
Const colVal = 2       ' values
Const rowBegin = 10    ' row where data begin
Const colForecast = 3  ' forecast

'
' initialise forecasting engine
'
Sub InitEngine()

    On Error GoTo wrong

    Dim shData As Object
    Set shData = ThisWorkbook.Sheets("Data")
    If shData.ComboSolver.ListCount > 0 Then Exit Sub

    Dim Aura As New AuraExpert

    ' fill list of solvers
    shData.ComboSolver.Clear
    Dim NumSolvers, i As Integer
    NumSolvers = Aura.SolversCount
    For i = 0 To NumSolvers - 1
        Rem If Aura.MinForecastLen(SolverName) > 0 Then
            shData.ComboSolver.AddItem Aura.SolverName(i)
        Rem End If
    Next i
    shData.ComboSolver.ListIndex = 0

    Exit Sub

wrong:
    MsgBox "Error communicating forecast engine!"

End Sub

'
' calculate forecasts
'
Sub CalculateForecasts()

    On Error GoTo wrong

    Dim shData As Object
    Set shData = ThisWorkbook.Worksheets("Data")

    ' calculate and fill forecasts
    Dim Aura As New AuraExpert
    Dim Predictor, NumInputs, InputLen, NumOutputs, ForecastLen, i, j As Integer
    Dim SolverName As String
    Predictor = 0
    SolverName = shData.ComboSolver.Text
    If Aura.MinForecastLen(SolverName) > 0 Then
        Predictor = 1
    End If

    ' dimensions
    NumInputs = 5
    InputLen = SeriesLen()
    NumOutputs = 1
    ForecastLen = 1
    Dim InputData() As Double
    Dim OutputData() As Double, VarianceData() As Double, DateData() As Date
    Dim SeriesNames As String, ModelBuffer As String, ModelParam As String
```



# ForeStock

```
ReDim InputData(InputLen * NumInputs)
ReDim DateData(InputLen)

For i = 1 To InputLen
    For j = 0 To NumInputs - 1
        InputData(i + j * InputLen) = shData.Cells(i + rowBegin, j + 2)
    Next j
Next i

'Aura = CreateObject("Aura.Expert")
'Aura.Calculate SolverName, NumInputs, InputLen, _
'                NumOutputs, ForecastLen, OutputData, VarianceData, DateData

SeriesNames = "Open" & vbLf & "High" & vbLf & "Low" & vbLf & "Close" & vbLf & "Volume"

Aura.CalculateForecasts SolverName, NumInputs, InputLen, InputData, _
                        NumOutputs, ForecastLen, OutputData, VarianceData, DateData, _
                        SeriesNames, ModelBuffer, ModelParam

For i = 0 To InputLen + Predictor - 1
    shData.Cells(i + rowBegin, 7) = OutputData(i)
Next i

Exit Sub

wrong:
    MsgBox "Error in calculation!"

End Sub
```

## Algorithm settings and limits

### Predictors

Algorithm	Input length		Forecast length		Number of inputs		Number of outputs	
	Min	Max	Min	Max	Min	Max	Min	Max
ARIMA	30	2147483647	1	2147483647	1	1	1	1
Exponential Fit	3	2147483647	1	32767	1	1	1	1
Finite Impulse Response NN	64	2147483647	1	2147483647	1	1	1	1
Finite State Markov Automaton	256	2147483647	1	1	1	1	1	1
Forecast with average value	1	2147483647	1	32767	1	1	1	1
History Prophet	1	2147483647	1	1	1	2147483647	1	2147483647
Linear Regression	7	2147483647	1	1	1	1	1	1
Linear regression	3	2147483647	1	2147483647	1	1	1	1
Logarithmic Fit	3	2147483647	1	2147483647	1	1	1	1
Naive Predictor	1	2147483647	1	1	1	2147483647	1	2147483647
Square Fit	3	2147483647	1	2147483647	1	1	1	1
Square Root Fit	3	2147483647	1	2147483647	1	1	1	1
Stepwise Best Regression MVAR	7	2147483647	1	1	2	2147483647	1	1

### Indicators

Algorithm	Input length		Forecast length		Number of inputs		Number of outputs	
	Min	Max	Min	Max	Min	Max	Min	Max
1 day ROC of 3 Smooth EMA	7	2147483647	0	0	1	1	1	1
Absolute Price Oscillator	7	2147483647	0	0	1	1	1	1
Aroon Down	7	2147483647	0	0	4	4	1	1
Aroon Oscillator	7	2147483647	0	0	4	4	1	1
Aroon Up	7	2147483647	0	0	4	4	1	1

# ForeStock

Average Price	7	2147483647	0	0	4	4	1	1
Average True Range	7	2147483647	0	0	4	4	1	1
Avg Direct Move Idx Rating	7	2147483647	0	0	4	4	1	1
Avg Directional Move Idx	7	2147483647	0	0	4	4	1	1
Chaikin AD Line	7	2147483647	0	0	5	5	1	1
Chaikin AD Oscillator	7	2147483647	0	0	5	5	1	1
Commodity Channel Index	7	2147483647	0	0	4	4	1	1
Directional Movement Index	7	2147483647	0	0	4	4	1	1
Double Exponential MA	7	2147483647	0	0	1	1	1	1
Exponential Moving Average	7	2147483647	0	0	1	1	1	1
FXS Adaptive Moving Average	7	2147483647	0	0	1	1	1	1
Hilbert Dominant Cycle Period	7	2147483647	0	0	1	1	1	1
Hilbert Dominant Cycle Phase	7	2147483647	0	0	1	1	1	1
Hilbert Phasor Components	7	2147483647	0	0	1	1	1	1
Hilbert SineWave	7	2147483647	0	0	1	1	1	1
Hilbert Transform Trendline	7	2147483647	0	0	1	1	1	1
Hilbert Trend vs Cycle Mode	7	2147483647	0	0	1	1	1	1
Kaufman Adaptive MA	7	2147483647	0	0	1	1	1	1
Linear Regr Angle	7	2147483647	0	0	1	1	1	1
Linear Regr Intercept	7	2147483647	0	0	1	1	1	1
Linear Regr Slope	7	2147483647	0	0	1	1	1	1
Linear Regression	7	2147483647	0	0	1	1	1	1
MESA Adaptive Moving Average	7	2147483647	0	0	1	1	1	1
Median Price	7	2147483647	0	0	4	4	1	1

# ForeStock

MidPoint over period	7	2147483647	0	0	1	1	1	1
Midpoint Price over period	7	2147483647	0	0	4	4	1	1
Minus Directional Indicator	7	2147483647	0	0	4	4	1	1
Minus Directional Movement	7	2147483647	0	0	4	4	1	1
Momentum	7	2147483647	0	0	1	1	1	1
Money Flow Index	7	2147483647	0	0	5	5	1	1
On Balance Volume	7	2147483647	0	0	5	5	1	1
Parabolic SAR	7	2147483647	0	0	4	4	1	1
Percentage Price Oscillator	7	2147483647	0	0	1	1	1	1
Plus Directional Indicator	7	2147483647	0	0	4	4	1	1
Plus Directional Movement	7	2147483647	0	0	4	4	1	1
Rate of change	7	2147483647	0	0	1	1	1	1
Rate of change Percentage	7	2147483647	0	0	1	1	1	1
Rate of change ratio	7	2147483647	0	0	1	1	1	1
Relative Strength Index	7	2147483647	0	0	1	1	1	1
Simple Moving Average	7	2147483647	0	0	1	1	1	1
Time Series Forecast	7	2147483647	0	0	1	1	1	1
Triangular Moving Average	7	2147483647	0	0	1	1	1	1
Triple EMA	7	2147483647	0	0	1	1	1	1
Triple EMA T3	7	2147483647	0	0	1	1	1	1
True Range	7	2147483647	0	0	4	4	1	1
Typical Price	7	2147483647	0	0	4	4	1	1
Weighted Close Price	7	2147483647	0	0	4	4	1	1
Weighted Moving Average	7	2147483647	0	0	1	1	1	1
Williams Percent R	7	2147483647	0	0	4	4	1	1

## References

---

1. ForeStock official web site <http://forestock.com>