QuantShare

Table of contents

1. Quantshare
  1.1 Introduction
  1.2 Tutorial
    1.2.1 Application
      1.2.1.1 Using the software
      1.2.1.2 Docking windows
      1.2.1.3 Toolbars
      1.2.1.4 Layouts
      1.2.1.5 Templates
      1.2.1.6 Workspaces
      1.2.1.7 Events
      1.2.1.8 Scripting
    1.2.2 Charting
      1.2.2.1 Charts
      1.2.2.2 Drawing tools
      1.2.2.3 Auto drawing tools
    1.2.3 Symbols
      1.2.3.1 Symbols
      1.2.3.2 Symbols Selection
    1.2.4 Data
      1.2.4.1 Databases
      1.2.4.2 Application Objects
    1.2.5 QuantShare Language
      1.2.5.1 QuantShare Language
      1.2.5.2 Advanced Rules
    1.2.6 Plug-ins
      1.2.6.1 Indicators
      1.2.6.2 Custom functions
      1.2.6.3 Composite
      1.2.6.4 Watch list
      1.2.6.5 Script Manager
      1.2.6.6 Widget Panel
      1.2.6.7 Sharing Server
      1.2.6.8 Divers
      1.2.6.9 Data
        1.2.6.9.1 ASCII Importer
        1.2.6.9.2 Downloader
        1.2.6.9.3 Data Viewer
      1.2.6.10 Analysis
        1.2.6.10.1 Rules Manager
        1.2.6.10.2 Ranking System Manager
        1.2.6.10.3 Simulator
        1.2.6.10.4 Advanced Money Management
        1.2.6.10.5 Screener
      1.2.6.11 Artificial Intelligence
        1.2.6.11.1 Artificial Intelligence

1.2.6.11.2 Optimizer
1.2.6.12 External
1.2.6.12.1 Portfolio123
1.3 QuantShare Language
1.3.1 Date-Time
1.3.1.1 Year
1.3.1.2 Date
1.3.1.3 DateTicks
1.3.1.4 Day
1.3.1.5 DayOfWeek
1.3.1.6 DayOfYear
1.3.1.7 Hour
1.3.1.8 Interval
1.3.1.9 Minute
1.3.1.10 Month
1.3.1.11 NbDays
1.3.1.12 Now
1.3.1.13 Second
1.3.1.14 TimeTicks
1.3.1.15 Week
1.3.2 Application Info
1.3.2.1 NbGroups
1.3.2.2 NbIndexes
1.3.2.3 NbIndustries
1.3.2.4 NbInGroup
1.3.2.5 NbInIndex
1.3.2.6 NbInIndustry
1.3.2.7 NbInMarket
1.3.2.8 NbInSector
1.3.2.9 NbMarkets
1.3.2.10 NbSectors
1.3.3 Candlestick Pattern
1.3.3.1 Cdl2crows 0
1.3.3.2 Cdl2crows 1
1.3.3.3 Cdl3blackcrows 0
1.3.3.4 Cdl3blackcrows 1
1.3.3.5 Cdl3inside 0
1.3.3.6 Cdl3inside 1
1.3.3.7 Cdl3linestrike 0
1.3.3.8 Cdl3linestrike 1
1.3.3.9 Cdl3outside 0
1.3.3.10 Cdl3outside 1
1.3.3.11 Cdl3staRsinsouth 0
1.3.3.12 Cdl3staRsinsouth 1
1.3.3.13 Cdl3whitesoldiers 0
1.3.3.14 Cdl3whitesoldiers 1
1.3.3.15 CdlAbandonedbaby 0
1.3.3.16 CdlAbandonedbaby 1
1.3.3.17 CdlAdvanceblock 0

1.3.3.18 CdlAdvanceblock 1
1.3.3.19 CdlBelthold 0
1.3.3.20 CdlBelthold 1
1.3.3.21 CdlBreakaway 0
1.3.3.22 CdlBreakaway 1
1.3.3.23 CdlClosingmarubozu 0
1.3.3.24 CdlClosingmarubozu 1
1.3.3.25 CdlConcealbabyswall 0
1.3.3.26 CdlConcealbabyswall 1
1.3.3.27 CdlCounterattack 0
1.3.3.28 CdlCounterattack 1
1.3.3.29 CdlDarkcloudcover 0
1.3.3.30 CdlDarkcloudcover 1
1.3.3.31 CdlDoji 0
1.3.3.32 CdlDoji 1
1.3.3.33 CdlDojistar 0
1.3.3.34 CdlDojistar 1
1.3.3.35 CdlDragonflydoji 0
1.3.3.36 CdlDragonflydoji 1
1.3.3.37 CdlEngulfing 0
1.3.3.38 CdlEngulfing 1
1.3.3.39 CdlEveningdojistar 0
1.3.3.40 CdlEveningdojistar 1
1.3.3.41 CdlEveningstar 0
1.3.3.42 CdlEveningstar 1
1.3.3.43 CdlGapsidesidewhite 0
1.3.3.44 CdlGapsidesidewhite 1
1.3.3.45 CdlGravestonedoji 0
1.3.3.46 CdlGravestonedoji 1
1.3.3.47 CdlHammer 0
1.3.3.48 CdlHammer 1
1.3.3.49 CdlHangingman 0
1.3.3.50 CdlHangingman 1
1.3.3.51 CdlHarami 0
1.3.3.52 CdlHarami 1
1.3.3.53 CdlHaramicross 0
1.3.3.54 CdlHaramicross 1
1.3.3.55 CdlHighwave 0
1.3.3.56 CdlHighwave 1
1.3.3.57 CdlHikkake 0
1.3.3.58 CdlHikkake 1
1.3.3.59 CdlHikkakemod 0
1.3.3.60 CdlHikkakemod 1
1.3.3.61 CdlHomingpigeon 0
1.3.3.62 CdlHomingpigeon 1
1.3.3.63 CdlIdentical3crows 0
1.3.3.64 CdlIdentical3crows 1
1.3.3.65 CdlInneck 0
1.3.3.66 CdlInneck 1

1.3.3.67 CdlInvertedhammer 0

1.3.3.68 CdlInvertedhammer 1

1.3.3.69 CdlKicking 0

1.3.3.70 CdlKicking 1

1.3.3.71 CdlKickingbylength 0

1.3.3.72 CdlKickingbylength 1

1.3.3.73 CdlLadderbottom 0

1.3.3.74 CdlLadderbottom 1

1.3.3.75 CdlLongleggeddoji 0

1.3.3.76 CdlLongleggeddoji 1

1.3.3.77 CdlLongline 0

1.3.3.78 CdlLongline 1

1.3.3.79 CdlMarubozu 0

1.3.3.80 CdlMarubozu 1

1.3.3.81 CdlMatchinglow 0

1.3.3.82 CdlMatchinglow 1

1.3.3.83 CdlMathold 0

1.3.3.84 CdlMathold 1

1.3.3.85 CdlMorningdojistar 0

1.3.3.86 CdlMorningdojistar 1

1.3.3.87 CdlMorningstar 0

1.3.3.88 CdlMorningstar 1

1.3.3.89 CdlOnneck 0

1.3.3.90 CdlOnneck 1

1.3.3.91 CdlPiercing 0

1.3.3.92 CdlPiercing 1

1.3.3.93 CdlRickshawman 0

1.3.3.94 CdlRickshawman 1

1.3.3.95 CdlRisefall3methods 0

1.3.3.96 CdlRisefall3methods 1

1.3.3.97 CdlSeparatinglines 0

1.3.3.98 CdlSeparatinglines 1

1.3.3.99 CdlShootingstar 0

1.3.3.100 CdlShootingstar 1

1.3.3.101 CdlShortline 0

1.3.3.102 CdlShortline 1

1.3.3.103 CdlSpinningtop 0

1.3.3.104 CdlSpinningtop 1

1.3.3.105 CdlStalledpattern 0

1.3.3.106 CdlStalledpattern 1

1.3.3.107 CdlSticksandwich 0

1.3.3.108 CdlSticksandwich 1

1.3.3.109 CdlTakuri 0

1.3.3.110 CdlTakuri 1

1.3.3.111 CdlTasukigap 0

1.3.3.112 CdlTasukigap 1

1.3.3.113 CdlThrusting 0

1.3.3.114 CdlThrusting 1

1.3.3.115 CdlTristar 0

- 1.3.3.116 CdlTristar 1
- 1.3.3.117 CdlUnique3river 0
- 1.3.3.118 CdlUnique3river 1
- 1.3.3.119 CdlUpsidegap2crows 0
- 1.3.3.120 CdlUpsidegap2crows 1
- 1.3.3.121 CdlXsidegap3methods 0
- 1.3.3.122 CdlXsidegap3methods 1

1.3.4 Charting
- 1.3.4.1 Plot
- 1.3.4.2 Plot1
- 1.3.4.3 PlotCandleStick
- 1.3.4.4 PlotCandleStick1
- 1.3.4.5 PlotFixed
- 1.3.4.6 PlotSymbol
- 1.3.4.7 Print
- 1.3.4.8 PrintChart
- 1.3.4.9 SetMaxScale
- 1.3.4.10 SetMinScale
- 1.3.4.11 SetScale
- 1.3.4.12 UpdateColor

1.3.5 Columns
- 1.3.5.1 AddColumn 0
- 1.3.5.2 AddColumn 1

1.3.6 Database
- 1.3.6.1 GetData
- 1.3.6.2 GetDataCount
- 1.3.6.3 GetDataCountInside
- 1.3.6.4 GetDataExtern
- 1.3.6.5 GetDataIndex
- 1.3.6.6 GetDataIndexDP
- 1.3.6.7 GetDataInside
- 1.3.6.8 GetDataStringCount
- 1.3.6.9 GetDataStringCountInside
- 1.3.6.10 IsDataExists

1.3.7 Divers
- 1.3.7.1 Output
- 1.3.7.2 OutputList
- 1.3.7.3 States
- 1.3.7.4 Ticker
- 1.3.7.5 Ticker1

1.3.8 Indicators
- 1.3.8.1 Ad 0
- 1.3.8.2 Ad 1
- 1.3.8.3 AdOsc 0
- 1.3.8.4 AdOsc 1
- 1.3.8.5 Adx 0
- 1.3.8.6 Adx 1
- 1.3.8.7 Adxr 0
- 1.3.8.8 Adxr 1

1.3.8.9 Apo 0
1.3.8.10 Apo 1
1.3.8.11 AroonDown 0
1.3.8.12 AroonDown 1
1.3.8.13 AroonOsc 0
1.3.8.14 AroonOsc 1
1.3.8.15 AroonUp 0
1.3.8.16 AroonUp 1
1.3.8.17 Atr 0
1.3.8.18 Atr 1
1.3.8.19 AvgPrice 0
1.3.8.20 AvgPrice 1
1.3.8.21 BbandSlower 0
1.3.8.22 BbandSlower 1
1.3.8.23 BbandsMiddle 0
1.3.8.24 BbandsMiddle 1
1.3.8.25 BbandsUpper 0
1.3.8.26 BbandsUpper 1
1.3.8.27 Bop 0
1.3.8.28 Bop 1
1.3.8.29 Cci 0
1.3.8.30 Cci 1
1.3.8.31 Cmo 0
1.3.8.32 Cmo 1
1.3.8.33 Correl 0
1.3.8.34 Correl 1
1.3.8.35 Dema 0
1.3.8.36 Dema 1
1.3.8.37 Dx 0
1.3.8.38 Dx 1
1.3.8.39 Ema 0
1.3.8.40 Ema 1
1.3.8.41 Fama 0
1.3.8.42 Fama 1
1.3.8.43 GapDown
1.3.8.44 GapUp
1.3.8.45 Hhv 0
1.3.8.46 Hhv 1
1.3.8.47 Ht_Dcperiod 0
1.3.8.48 Ht_Dcperiod 1
1.3.8.49 Ht_Dcphase 0
1.3.8.50 Ht_Dcphase 1
1.3.8.51 Ht_Leadsine 0
1.3.8.52 Ht_Leadsine 1
1.3.8.53 Ht_Phasorphase 0
1.3.8.54 Ht_Phasorphase 1
1.3.8.55 Ht_Phasorquadrature 0
1.3.8.56 Ht_Phasorquadrature 1
1.3.8.57 Ht_Sine 0

1.3.8.58 Ht_Sine 1
1.3.8.59 Ht_Trendline 0
1.3.8.60 Ht_Trendline 1
1.3.8.61 Ht_Trendmode 0
1.3.8.62 Ht_Trendmode 1
1.3.8.63 Inside
1.3.8.64 Kama 0
1.3.8.65 Kama 1
1.3.8.66 LinearReg 0
1.3.8.67 LinearReg 1
1.3.8.68 LinearReg_Angle 0
1.3.8.69 LinearReg_Angle 1
1.3.8.70 LinearReg_Intercept 0
1.3.8.71 LinearReg_Intercept 1
1.3.8.72 LinearReg_Slope 0
1.3.8.73 LinearReg_Slope 1
1.3.8.74 Llv 0
1.3.8.75 Llv 1
1.3.8.76 Ma 0
1.3.8.77 Ma 1
1.3.8.78 Macd 0
1.3.8.79 Macd 1
1.3.8.80 MacdExt 0
1.3.8.81 MacdExt 1
1.3.8.82 MacdExtHist 0
1.3.8.83 MacdExtHist 1
1.3.8.84 MacdExtSignal 0
1.3.8.85 MacdExtSignal 1
1.3.8.86 MacdHist 0
1.3.8.87 MacdHist 1
1.3.8.88 MacdSignal 0
1.3.8.89 MacdSignal 1
1.3.8.90 Mama 0
1.3.8.91 Mama 1
1.3.8.92 MedPrice 0
1.3.8.93 MedPrice 1
1.3.8.94 Mfi 0
1.3.8.95 Mfi 1
1.3.8.96 MidPoint 0
1.3.8.97 MidPoint 1
1.3.8.98 MidPrice 0
1.3.8.99 MidPrice 1
1.3.8.100 Minus_Di 0
1.3.8.101 Minus_Di 1
1.3.8.102 Minus_Dm 0
1.3.8.103 Minus_Dm 1
1.3.8.104 Mom 0
1.3.8.105 Mom 1
1.3.8.106 Natr 0

1.3.8.107 Natr 1

1.3.8.108 Obv 0

1.3.8.109 Obv 1

1.3.8.110 Outside

1.3.8.111 Plus_Di 0

1.3.8.112 Plus_Di 1

1.3.8.113 Plus_Dm 0

1.3.8.114 Plus_Dm 1

1.3.8.115 Ppo 0

1.3.8.116 Ppo 1

1.3.8.117 Roc 0

1.3.8.118 Roc 1

1.3.8.119 Rocp 0

1.3.8.120 Rocp 1

1.3.8.121 Rocr 0

1.3.8.122 Rocr 1

1.3.8.123 Rocr100 0

1.3.8.124 Rocr100 1

1.3.8.125 Rsi 0

1.3.8.126 Rsi 1

1.3.8.127 Sar 0

1.3.8.128 Sar 1

1.3.8.129 SarExt 0

1.3.8.130 SarExt 1

1.3.8.131 Sma 0

1.3.8.132 Sma 1

1.3.8.133 Stddev 0

1.3.8.134 Stddev 1

1.3.8.135 StochFastd 0

1.3.8.136 StochFastd 1

1.3.8.137 StochFastk 0

1.3.8.138 StochFastk 1

1.3.8.139 StochRsiFastd 0

1.3.8.140 StochRsiFastd 1

1.3.8.141 StochRsiFastk 0

1.3.8.142 StochRsiFastk 1

1.3.8.143 StochSlowd 0

1.3.8.144 StochSlowd 1

1.3.8.145 StochSlowk 0

1.3.8.146 StochSlowk 1

1.3.8.147 T3 0

1.3.8.148 T3 1

1.3.8.149 Tema 0

1.3.8.150 Tema 1

1.3.8.151 Trange 0

1.3.8.152 Trange 1

1.3.8.153 Trima 0

1.3.8.154 Trima 1

1.3.8.155 Trix 0

1.3.8.156 Trix 1
1.3.8.157 Tsf 0
1.3.8.158 Tsf 1
1.3.8.159 TypPrice 0
1.3.8.160 TypPrice 1
1.3.8.161 UltOsc 0
1.3.8.162 UltOsc 1
1.3.8.163 Var 0
1.3.8.164 Var 1
1.3.8.165 WclPrice 0
1.3.8.166 WclPrice 1
1.3.8.167 Willr 0
1.3.8.168 Willr 1
1.3.8.169 Wma 0
1.3.8.170 Wma 1
1.3.9 Math
1.3.9.1 Abs
1.3.9.2 Atan
1.3.9.3 Avg 0
1.3.9.4 Avg 1
1.3.9.5 Ceil
1.3.9.6 Cos
1.3.9.7 Cosh
1.3.9.8 DivRem
1.3.9.9 Exp
1.3.9.10 Floor
1.3.9.11 Frac
1.3.9.12 IeeeRemainder
1.3.9.13 Int
1.3.9.14 Log
1.3.9.15 Log10
1.3.9.16 Max
1.3.9.17 Min
1.3.9.18 Pow
1.3.9.19 Random
1.3.9.20 Round
1.3.9.21 Sharpe
1.3.9.22 Sign
1.3.9.23 Sin
1.3.9.24 Sinh
1.3.9.25 Sqrt
1.3.9.26 Sum 0
1.3.9.27 Sum 1
1.3.9.28 Tan
1.3.9.29 Tanh
1.3.10 Optimal Signal
1.3.10.1 BSignal
1.3.10.2 BSignalLong
1.3.10.3 BSignalShort

- 1.3.10.4 EntryLongProfit
- 1.3.10.5 EntryLongProfitCond
- 1.3.10.6 EntryLongProfitCondExitRule
- 1.3.10.7 EntryLongProfitExitRule
- 1.3.10.8 EntryLongProfitPerBar
- 1.3.10.9 EntryLongProfitPerBarCond
- 1.3.10.10 EntryLongProfitPerBarCondExitRule
- 1.3.10.11 EntryLongProfitPerBarExitRule
- 1.3.10.12 EntryShortProfit
- 1.3.10.13 EntryShortProfitCond
- 1.3.10.14 EntryShortProfitCondExitRule
- 1.3.10.15 EntryShortProfitExitRule
- 1.3.10.16 EntryShortProfitPerBar
- 1.3.10.17 EntryShortProfitPerBarCond
- 1.3.10.18 EntryShortProfitPerBarCondExitRule
- 1.3.10.19 EntryShortProfitPerBarExitRule

1.3.11 Pre-calculated-value
- 1.3.11.1 GetPreCalculatedValue 0
- 1.3.11.2 GetPreCalculatedValue 1

1.3.12 Predictions
- 1.3.12.1 Predict
- 1.3.12.2 PredictTicker

1.3.13 Ranking
- 1.3.13.1 Ranking

1.3.14 Simulator
- 1.3.14.1 BuyPrice
- 1.3.14.2 CoverPrice
- 1.3.14.3 Drawdown
- 1.3.14.4 Equity
- 1.3.14.5 Optimize
- 1.3.14.6 SellPrice
- 1.3.14.7 SetSimCommission
- 1.3.14.8 SetSimLongRank
- 1.3.14.9 SetSimPeriods
- 1.3.14.10 SetSimRefSymbol
- 1.3.14.11 SetSimSetting
- 1.3.14.12 SetSimShortRank
- 1.3.14.13 SetSimStop
- 1.3.14.14 SetSimTiming
- 1.3.14.15 ShortPrice

1.3.15 String
- 1.3.15.1 GetDataString
- 1.3.15.2 GetDataStringIndex
- 1.3.15.3 GetDataStringInside
- 1.3.15.4 StringContains
- 1.3.15.5 StringEqual
- 1.3.15.6 StringExtract
- 1.3.15.7 StringExtractEnds
- 1.3.15.8 StringExtractStart

1.3.15.9 StringInsert
1.3.15.10 StringLength
1.3.15.11 StringReplace
1.3.16 Symbol Info
1.3.16.1 Address
1.3.16.2 Country
1.3.16.3 Currency
1.3.16.4 FullName
1.3.16.5 GetPreCalculatedValueString 0
1.3.16.6 GetPreCalculatedValueString 1
1.3.16.7 Group
1.3.16.8 Index
1.3.16.9 Industry
1.3.16.10 Market
1.3.16.11 Name
1.3.16.12 Name1
1.3.16.13 Name2
1.3.16.14 Name3
1.3.16.15 Sector
1.3.16.16 Website
1.3.17 Technical
1.3.17.1 AvgIf 0
1.3.17.2 AvgIf 1
1.3.17.3 BarsSince
1.3.17.4 Count
1.3.17.5 Cross
1.3.17.6 GFun
1.3.17.7 HhvLb 0
1.3.17.8 HhvLb 1
1.3.17.9 Iff
1.3.17.10 IsFalse
1.3.17.11 IsNa
1.3.17.12 IsNaN
1.3.17.13 IsNoNa
1.3.17.14 IsNoNaN
1.3.17.15 IsNoNull
1.3.17.16 IsNoZero
1.3.17.17 IsNull
1.3.17.18 IsTrue
1.3.17.19 LastNotNullValue
1.3.17.20 LastValue
1.3.17.21 LlvLb 0
1.3.17.22 LlvLb 1
1.3.17.23 NaNtoLast
1.3.17.24 NaNtoZero
1.3.17.25 Perf
1.3.17.26 PerfD
1.3.17.27 Rank
1.3.17.28 Ref

1.3.17.29 RefZero

1.3.17.30 RemoveSameSignals 0

1.3.17.31 RemoveSameSignals 1

1.3.17.32 RFun

1.3.17.33 SumIf 0

1.3.17.34 SumIf 1

1.3.17.35 Value

1.3.17.36 ValueWhen 0

1.3.17.37 ValueWhen 1

1.4 QuantShare CSharp & JScript.Net

1.4.1 Advanced Money Management

1.4.1.1 OnEndPeriod

1.4.1.2 OnEndSimulation

1.4.1.3 OnStartSimulation

1.4.1.4 OnClosePosition

1.4.1.5 OnNewPosition

1.4.1.6 Members

1.4.1.6.1 MMDivers

1.4.1.6.2 MMOnEndPeriod

1.4.1.6.3 MMOnEndSimulation

1.4.1.6.4 MMOptimize

1.4.1.6.5 MMPortfolio

1.4.1.6.6 MMTradingSystemSettings

1.4.1.6.7 MMVariables

1.4.1.6.8 MMOnNewPosition

1.4.1.6.9 MMExitPosition

1.4.1.6.10 MMNewPosition

1.4.1.6.11 MMOnClosePosition

1.4.1.6.12 MMPosition

1.4.1.6.13 TimeSeries

1.4.1.6.14 VectorCustom

1.4.1.6.15 CustomDatabaseValues

1.4.1.6.16 MMData

1.4.1.6.17 MMOrders

1.4.1.6.18 MMOnStartSimulation

1.4.1.6.19 MMParser

1.4.1.6.20 MMEntrySignals

1.4.1.6.21 MMExitSignals

1.4.1.6.22 _TradingOrder

1.4.2 Analyze Ranking System Metrics

1.4.2.1 Analyze Ranking System Metrics

1.4.2.2 Members

1.4.2.2.1 BucketDetails

1.4.2.2.2 BucketDetailsDate

1.4.2.2.3 RankingSystemPerformanceDetails

1.4.3 Analyze Rules Metrics

1.4.3.1 Analyze Rules Metrics

1.4.3.2 Members

1.4.3.2.1 PositionDetails

1.4.3.2.2 SymbolPositionDetails
1.4.3.2.3 RulesFunctions
1.4.4 Parser Post-Script
1.4.4.1 Downloader Post-Script
1.4.4.2 Members
1.4.4.2.1 PostScript
1.4.4.2.2 PostScriptRow
1.4.5 Parser Pre-Script
1.4.5.1 Downloader Pre-Script
1.4.5.2 Members
1.4.5.2.1 PreScript
1.4.5.2.2 PreScriptRow
1.4.6 Downloader URL-Script
1.4.6.1 Downloader URL-Script
1.4.6.2 Members
1.4.6.2.1 Net
1.4.6.2.2 URLScriptFunctions
1.4.6.2.3 HTMLElement
1.4.6.2.4 URLPath
1.4.7 Indicators
1.4.7.1 Indicators
1.4.7.2 Members
1.4.7.2.1 CFunctions
1.4.7.2.2 CustomDatabaseValues
1.4.7.2.3 QSFormula
1.4.7.2.4 VectorCustom
1.4.7.2.5 VectorD
1.4.7.2.6 VectorDate
1.4.7.2.7 VectorS
1.4.8 Perdiction Model Metrics
1.4.8.1 Perdiction Model Metrics
1.4.8.2 Members
1.4.8.2.1 PredictionErrors
1.4.9 Simulation Metrics
1.4.9.1 Simulation Metrics

1. Quantshare

QUANTSHARE *Trading software ⊕ Sharing server ⊕ Social network*

QuantShare is an advanced technical/fundamental analysis program.

QuantShare is a combination of trading software, sharing server and social network website.
We have built a flexible trading software whose possibilities are only limited by your imagination. But because we know that this is not enough, we have created a sharing server to let you exchange items with other traders in order to feed your imagination, increase your productivity, give you the opportunity to use advanced traders tools, and help you improve your trading skills.

QuantShare

[TOP]

1.1 Introduction

QUANTSHARE Trading software ⊕ Sharing server ⊕ Social network

QuantShare is an advanced technical/fundamental analysis program.

QuantShare is a combination of trading software, sharing server and social network website.
We have built a flexible trading software whose possibilities are only limited by your imagination. But because we know that this is not enough, we have created a sharing server to let you exchange items with other traders in order to feed your imagination, increase your productivity, give you the opportunity to use advanced traders tools, and help you improve your trading skills.

QuantShare

[TOP]

1.2 Tutorial

QUANTSHARE Trading software ⊕ Sharing server ⊕ Social network

This chapter will guide you through the tutorial parts:

Application:

Using the software
Docking windows
Toolbars
Layouts
Templates
Workspaces
Events

Charting:

Charts
Drawing tools
Auto drawing tools

Symbols:

Symbols
Symbols Selection

Data:

Databases
Application Objects

QuantShare Language:

QuantShare Language
FollowedBy

Plug-ins:

Indicators
Custom functions
Composite
Watch List
Script Manager
Widget Panel
Sharing Server
Divers

Rules Manager
Ranking System Manager
Simulator
Advanced Money Management
Screener

ASCII Importer
Downloader
Data Viewer

Artificial Intelligence Optimizer

Portfolio123

[TOP]

1.2.1 Application

1.2.1.1 Using the software

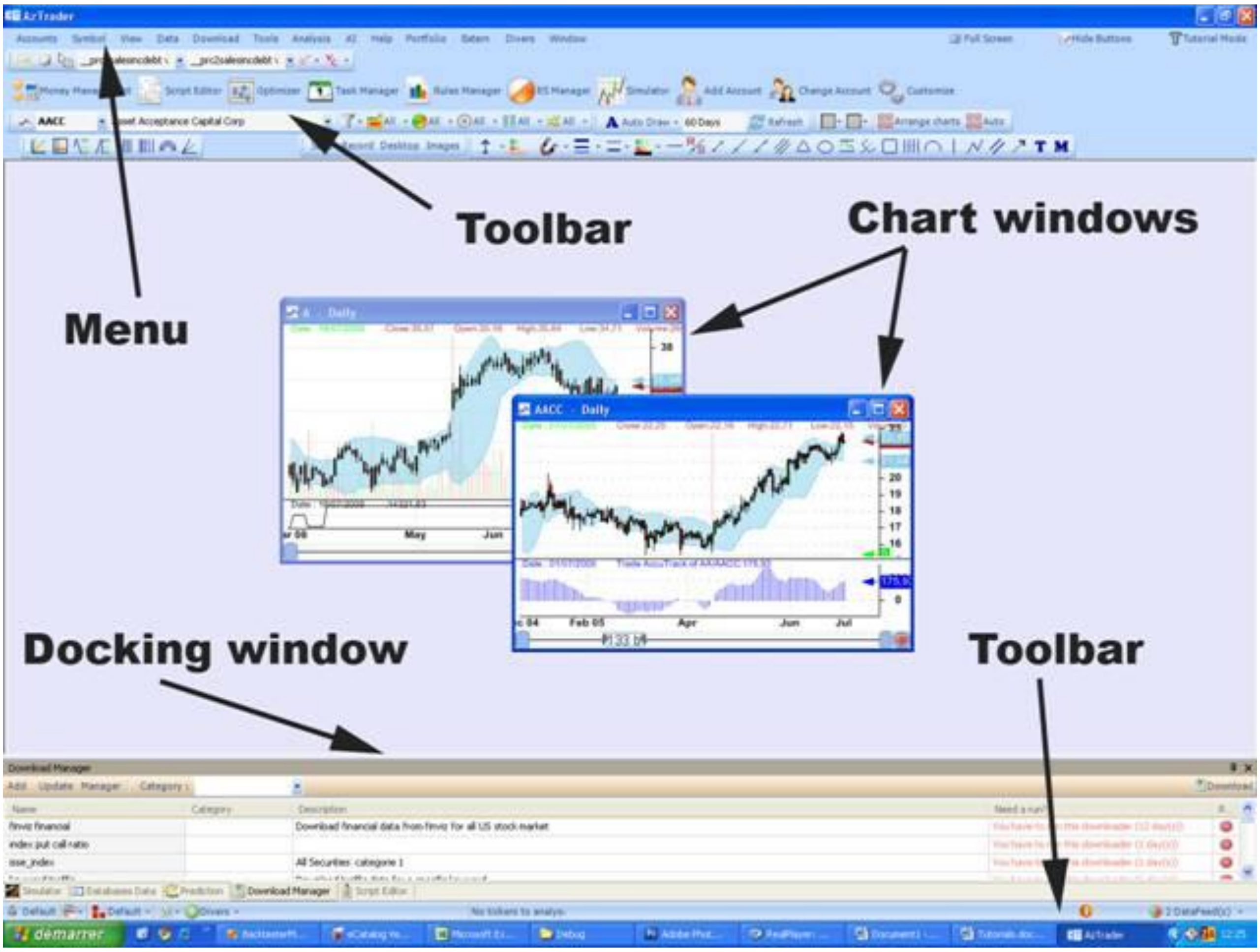
1. Using the software

1.1. Starting the program

To start the program, double-click the QuantShare icon on the Windows desktop or select QuantShare menu item from the Programs menu.

1.2. Main window

The main application window contains the menu, the toolbars, the docking windows and the chart windows.



1.3. Application settings

You can specify some application settings like the default windows color, the background color, the thread window settings...

To access the application settings (*Accounts -> Application Settings*)

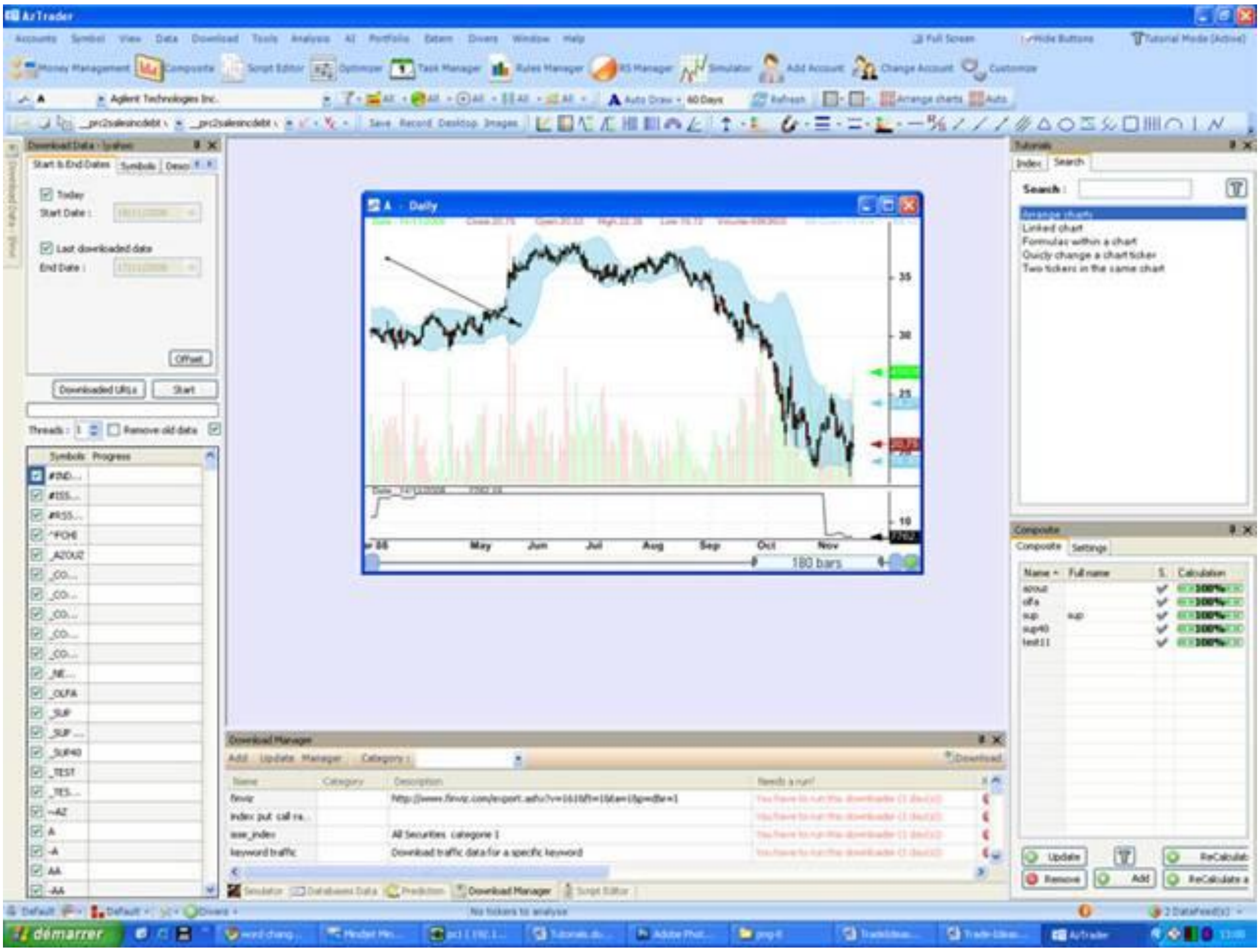
1.2.1.2 Docking windows

1. Docking windows

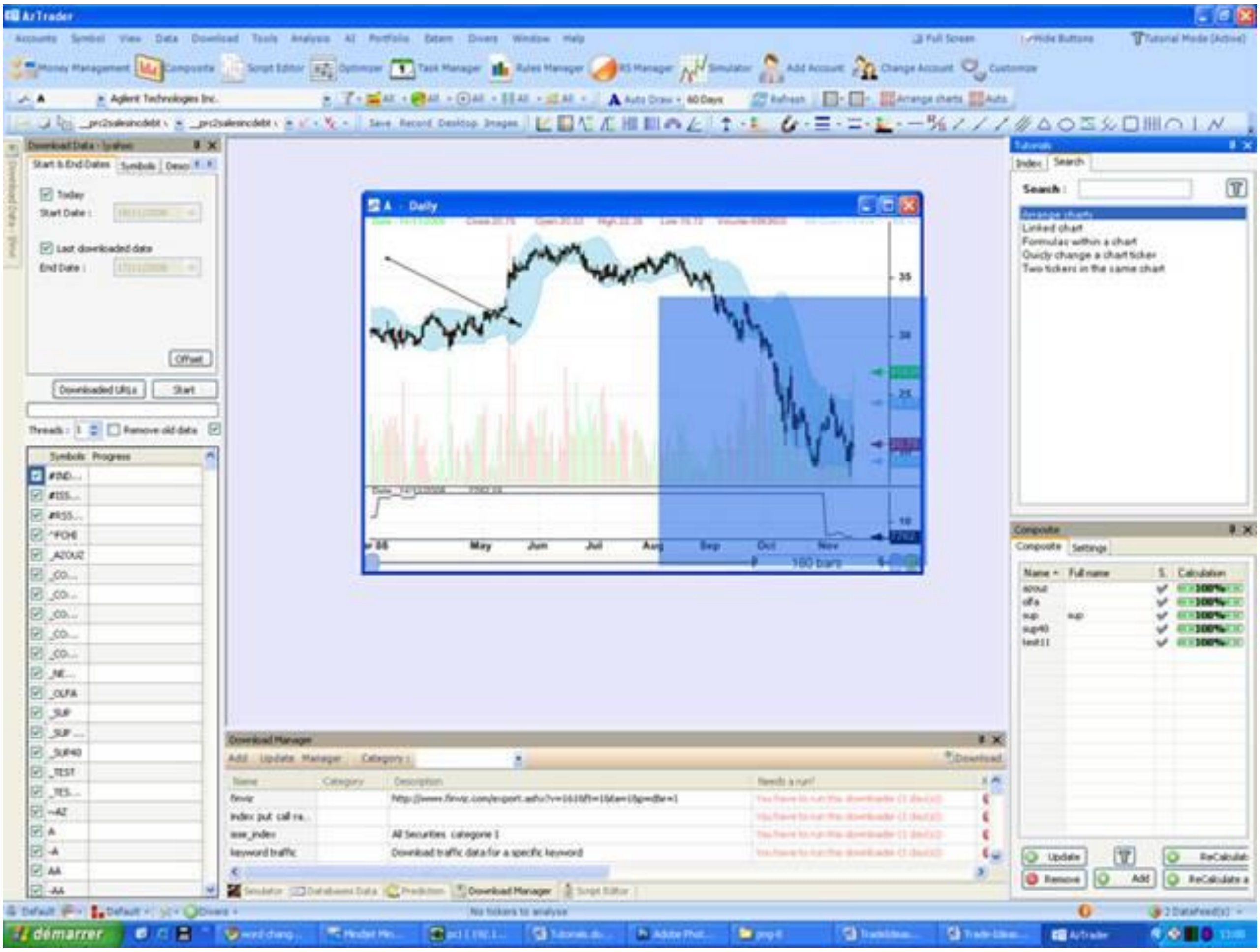
A docking window can stick to the main application boundaries or to the other windows boundaries.

1.1. Move a docking window

In the picture below, we have a chart window and four docking windows.
To move the right panel or window to the left, first click on the caption of the right panel (Where it is written Composite in this example).



Move the mouse while keeping the mouse left button pressed; a blue rectangle will appears, continue moving this rectangle to the left panel.

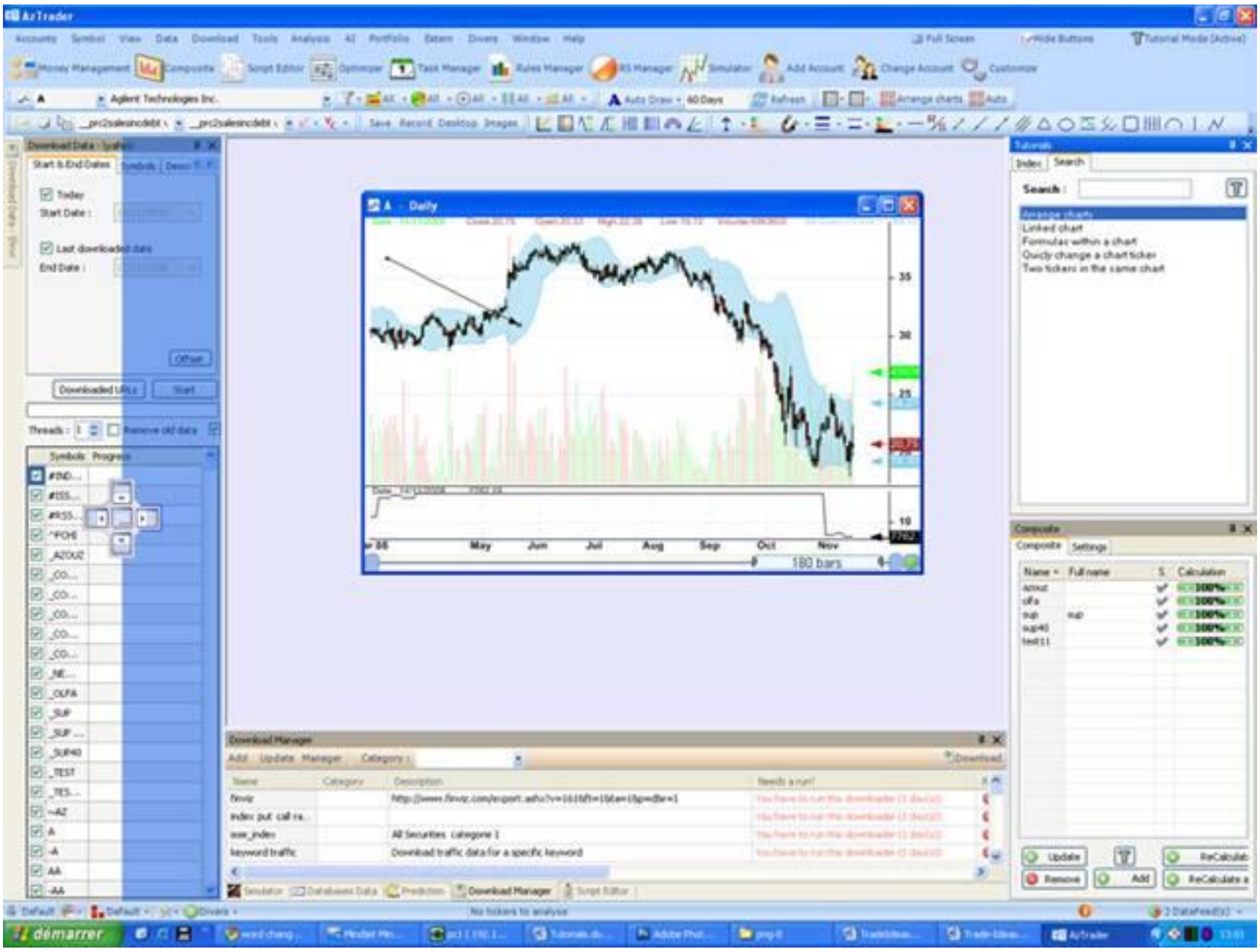


Arrange charts
Linked chart
Formulas within a chart
Quicly change a chart ticker
Two tickers in the same chart

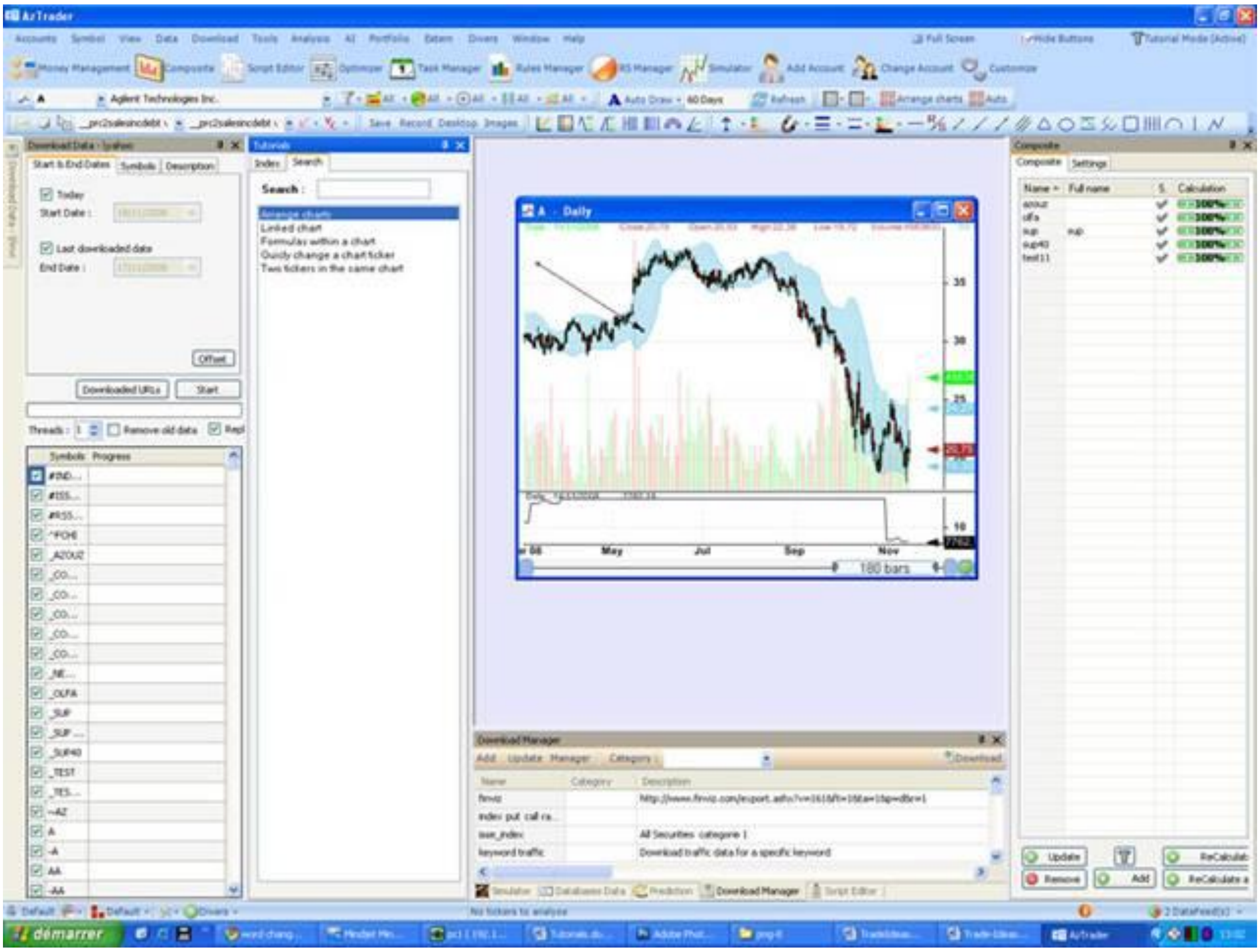
Name	Full name	S.	Calculation
azouz		✓	100%
olfa		✓	100%
sup	sup	✓	100%
sup40		✓	100%
test11		✓	100%

Name	Category	Description	Needs a run?
finviz		http://www.finviz.com/export.ashx?v=161&ft=1&ta=1&p=d&r=1	You have to run this downloader (1 day(s))
index put call ra...			You have to run this downloader (1 day(s))
asse_index		All Securities categorie 1	You have to run this downloader (1 day(s))
keyword traffic		Download traffic data for a specific keyword	You have to run this downloader (1 day(s))

An icon with five rectangles appears, select the right rectangle then release the mouse button.



Name	Category	Description	Needs a run?
finviz		http://www.finviz.com/export.ashx?v=161&f=1&ta=1&p=d&r=1	You have to run this downloader (1 day(s))
index put call ra...			You have to run this downloader (1 day(s))
isse_index		All Securities categorie 1	You have to run this downloader (1 day(s))
keyword traffic		Download traffic data for a specific keyword	You have to run this downloader (1 day(s))



Name	Category	Description
finviz		http://www.finviz.com/export.ashx?v=161&f=1&ta=1&p=d&r=1
index put call ra...		
isse_index		All Securities categorie 1
keyword traffic		Download traffic data for a specific keyword

1.2.1.3 Toolbars

1. Toolbars

This application contains two main toolbars, one located at the top below the menu and the other at the bottom.

1.1. Top Toolbar

The top toolbar contains mainly symbol's selection combo boxes and drawing tools buttons, it contains some other buttons created by the plug-ins and a list of shortcuts.

1.2. Bottom Toolbar

The bottom toolbar is used to display account name and to manage templates, layouts and workspaces.

The bottom toolbar has the following items:

- The current account name
- Templates menu
- Layouts menu
- Workspace menu
- Divers menu, which contains the list of shortcuts
- Symbols working information
- Message and alerts
- Intraday data-feeds

1.3. Accounts

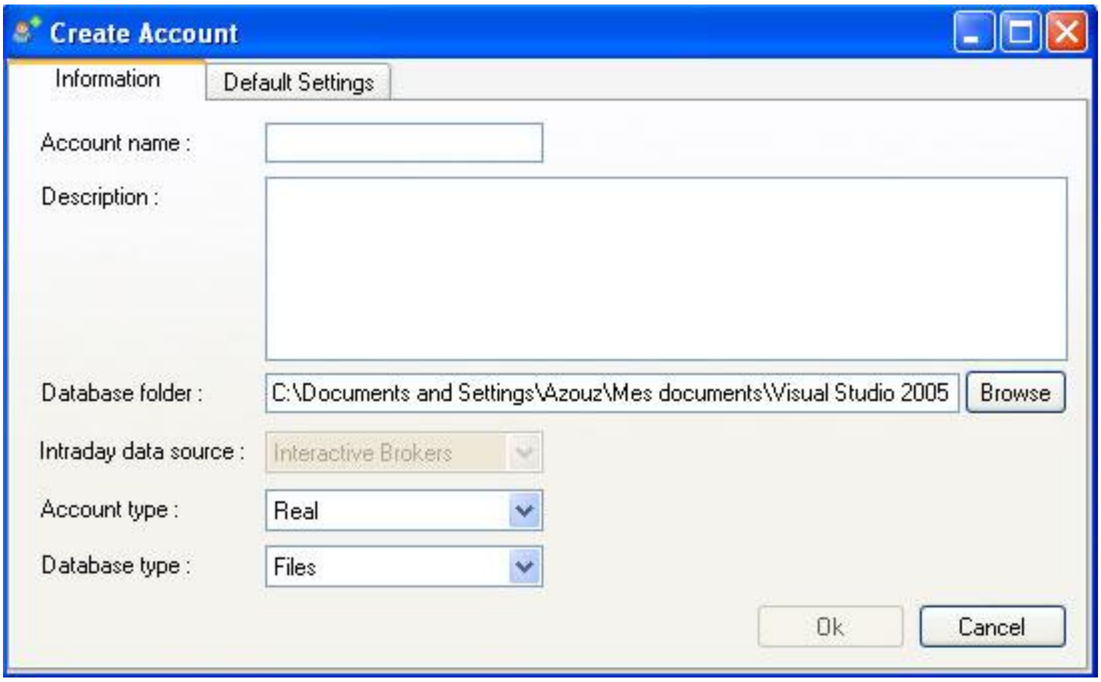
The application starts the first time with a default account name. You can add, remove and update accounts later. Each account has its own items, databases data, quotes…

1.3.1. Account files

Account files are located in the directory 'Databases'.
Each account has a sub-directory in the 'Databases' directory.
Example if your account name is 'Trader' then the full path to this account is 'Application Path\Databases\Trader', and within this directory are located all the objects, databases associated to this account.

1.3.2. Create an account

To create a new account (*Accounts -> Add Account*)



Account name :

Description :

Database folder : C:\Documents and Settings\Azouz\Mes documents\Visual Studio 2005

Intraday data source : Interactive Brokers

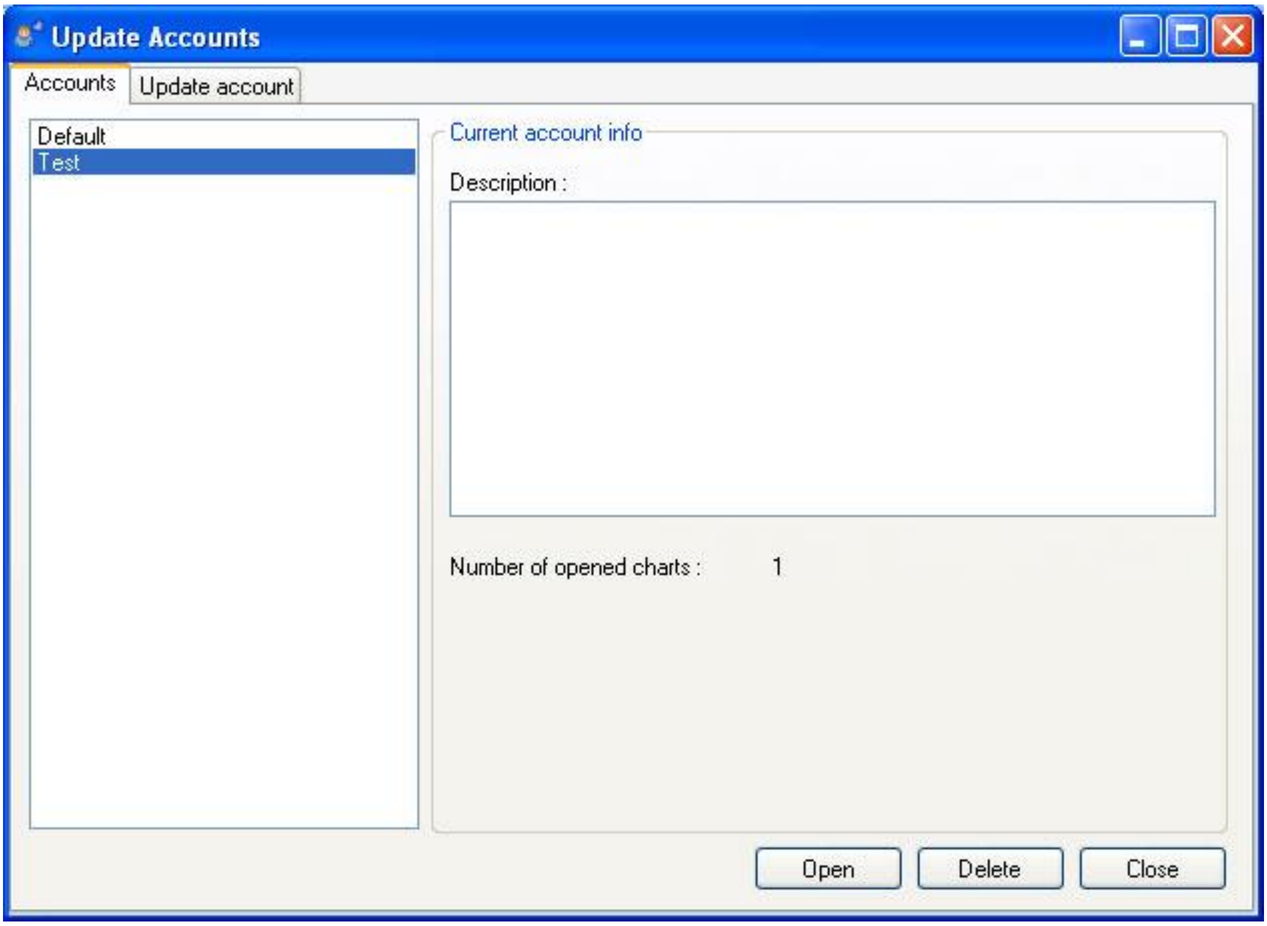
Account type : Real

Database type : Files

The database folder is updated automatically after you type the account name.

1.3.3. Change the current account

To change the current account (*Accounts -> Change Account*)



Default
Test

Current account info

Description :

Number of opened charts : 1

Select an account then click on 'Open'.

1.3.4. Update an account

To update an account (*Accounts -> Change Account*)

Select an account then select the 'Update account' tab.
Click on 'Update and reload' when you finish.

1.3.5. Remove an account

To remove an account, click on 'Change Account', select an account then click on 'Delete'.

1.3.6. Associate an existing database to an account

Select (*Accounts -> Change Account*)

Select an account then click on the 'Update account' tab.
In the database folder textbox, choose an existing directory that contains the data you want to use (usually the directory of another account).

1.2.1.4 Layouts

1. Layouts

1.1. Layouts

A layout contains information such as:

How many panes a chart have?
What is the formula on each pane?
What is the template associated with each pane?
...

1.2. Create a layout

To create a layout from a chart, right click on that chart and click on 'Save layout as...' type a layout name then saves.

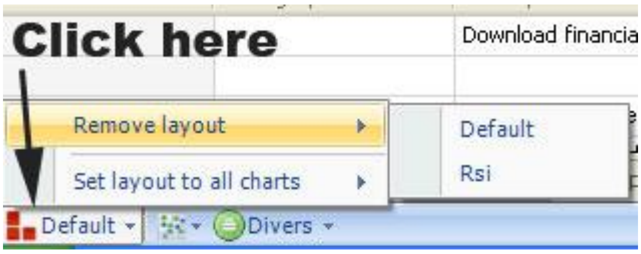
1.3. Default layout

The default layout is the layout that will be used when creating a new chart (*View -> New chart*).
To set a layout as a 'Default layout', right click on a chart that contains the layout you want to use as a default layout then click on 'Set current layout as default'.

NB: the default layout will be saved as 'Default.lay'. If this file is removed then the program will create a new default layout.

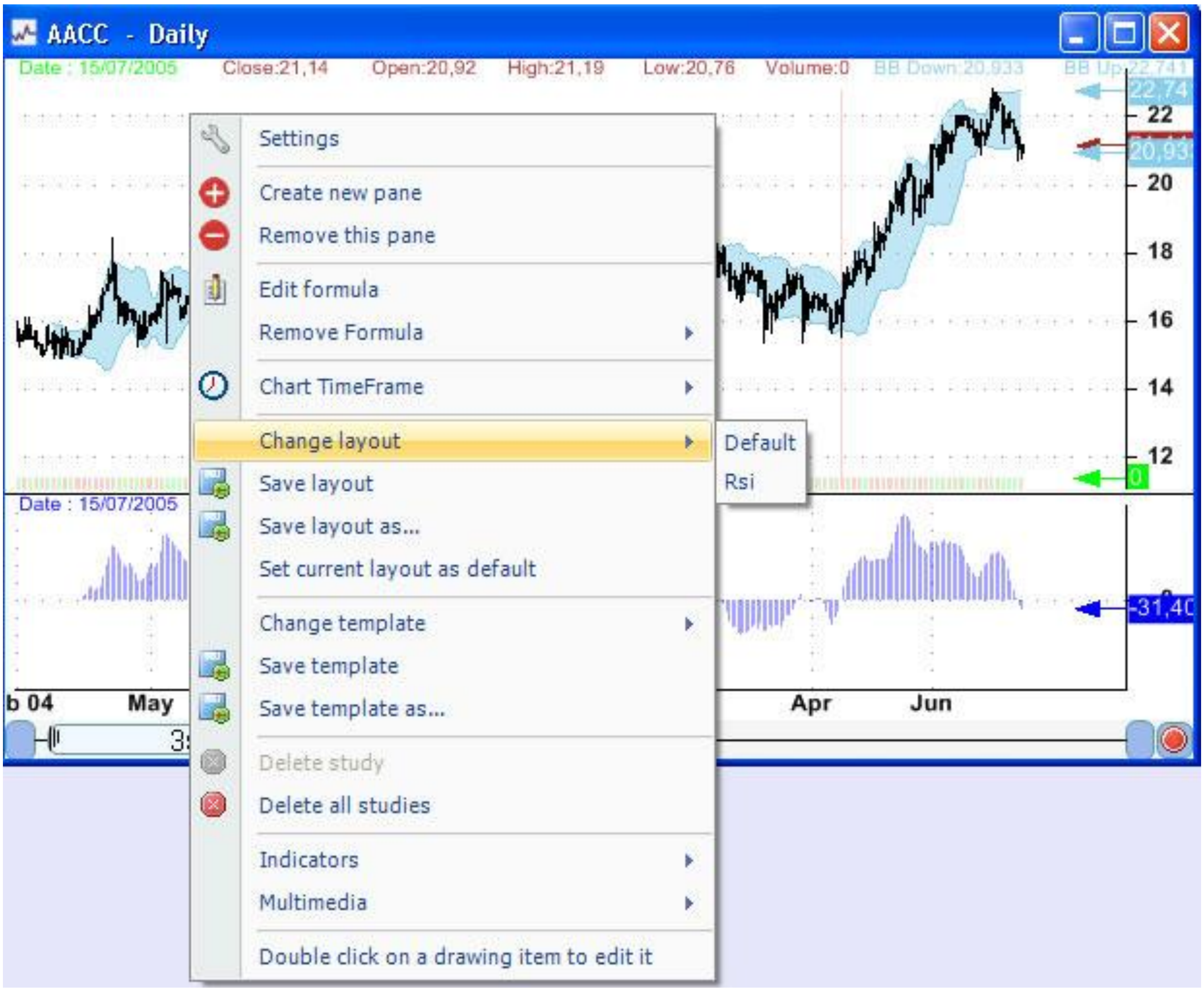
1.4. Remove a layout

Click on the button, in the toolbar, as shown in the picture, select 'Remove layout' and then choose the layout you want to remove.



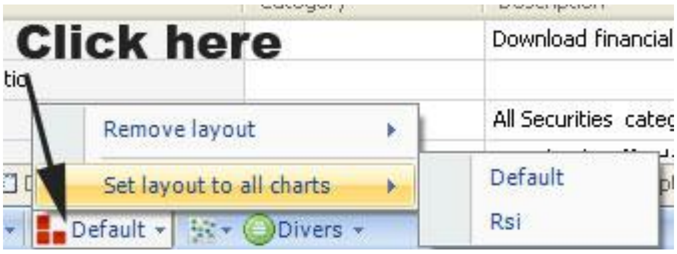
1.5. Update a chart layout

To update a chart's layout, right click on a chart, select 'Change layout' then selects the layout you want to use.



1.6. Set the same layout to all charts

To set a layout to all opened charts, click on the button, in the toolbar, as shown in the picture, select 'Set layout to all charts', and then choose a layout.



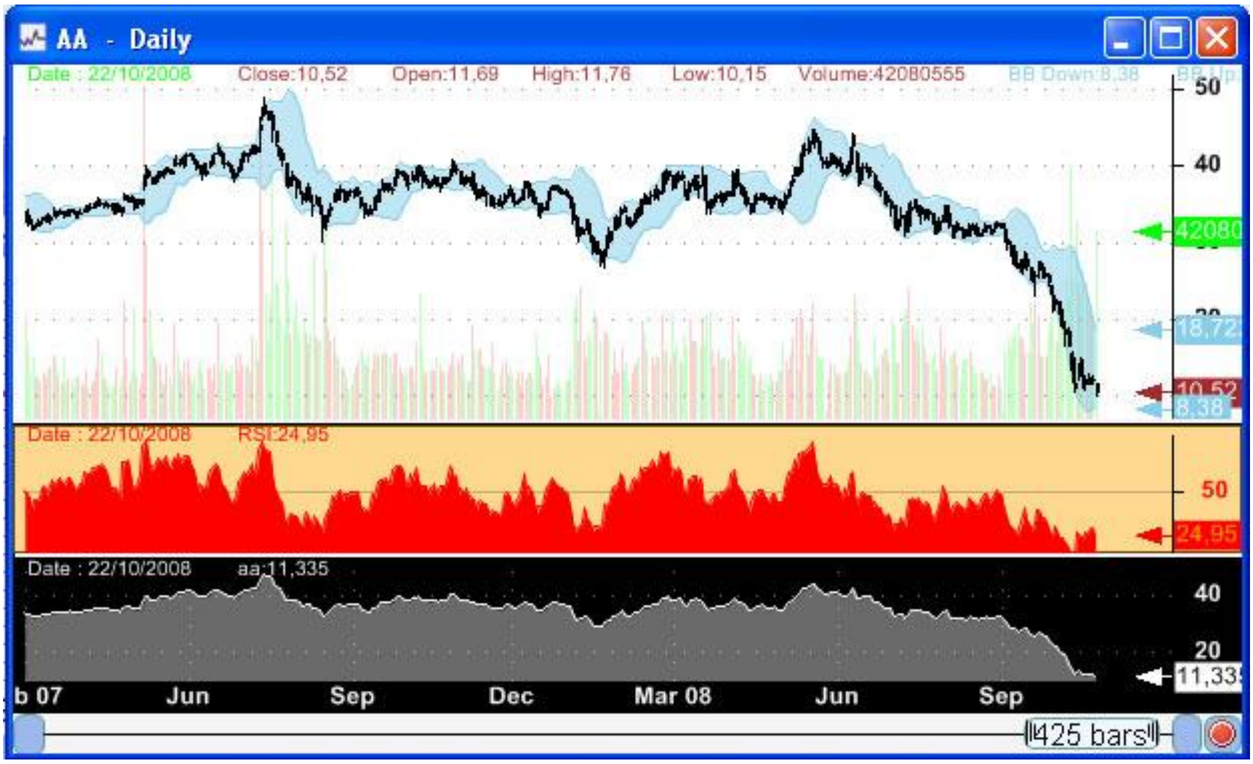
1.2.1.5 Templates

1. Templates

1.1. Templates

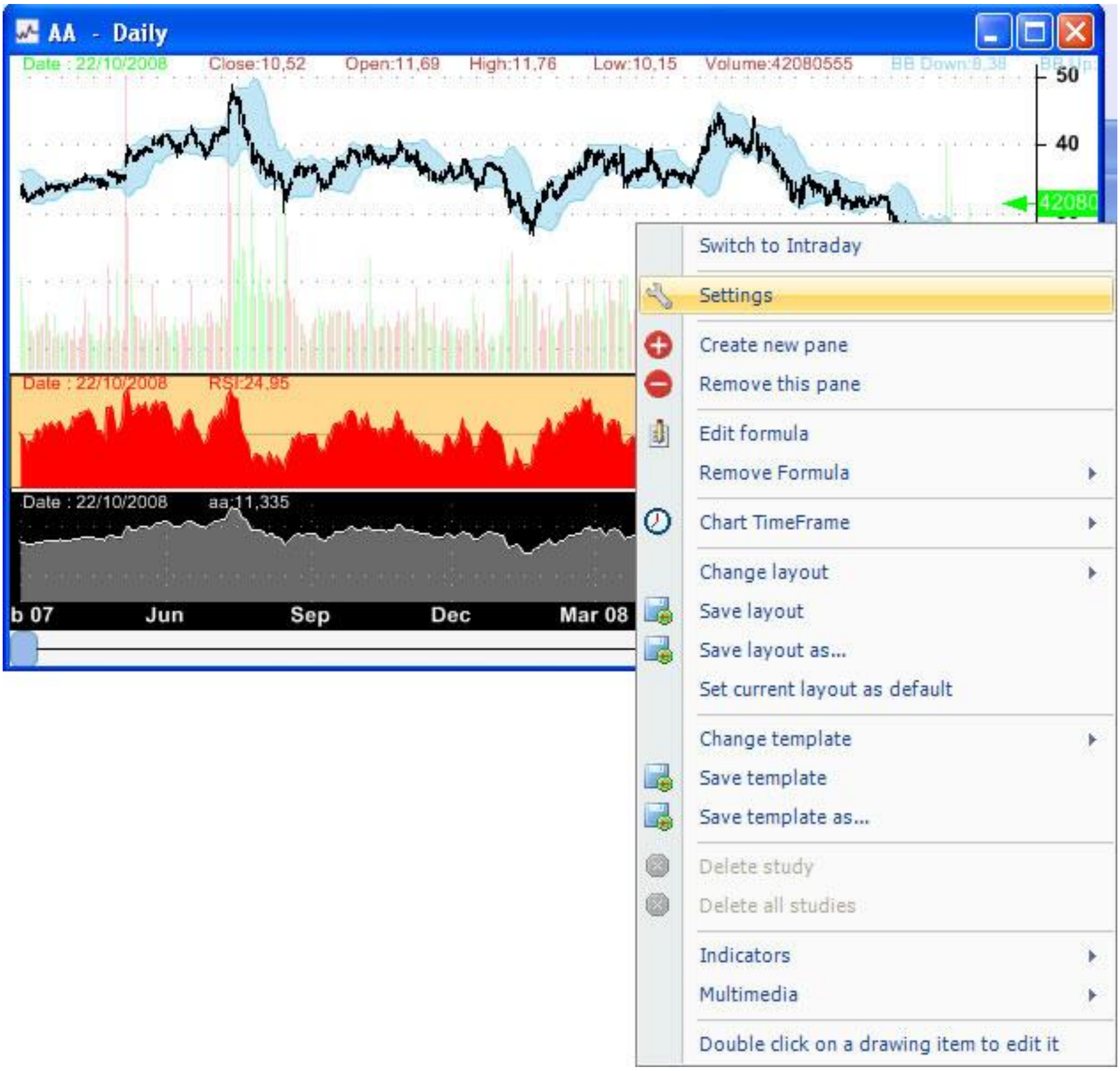
Each chart contains one or more panes. A template is a pane visual setting; it defines the pane's back color, grid line width, the axis colors...

You can easily create new templates, modify them, and associate them to other panes...
A chart can also contain several panes; each pane can have a different template.



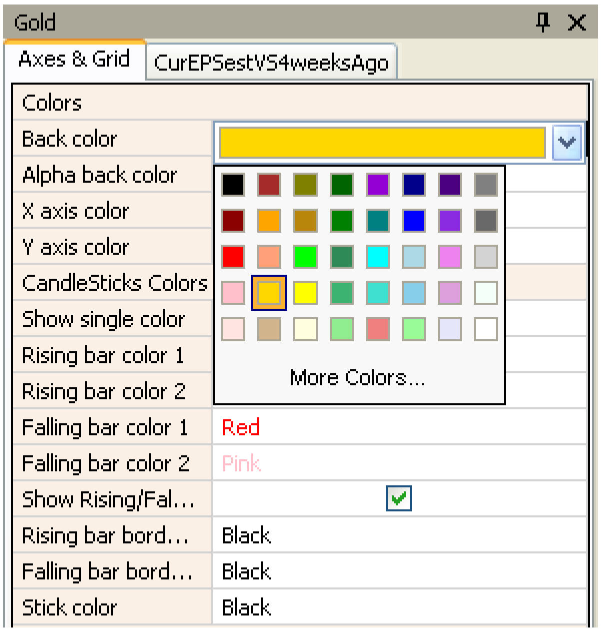
1.2. Create a template

On a chart's pane, right click then select 'Settings' in the menu.



In the settings form, you can update any field by clicking on its value.
When you finish, right click again on the chart's pane and click on 'Save template as...'

Type in the template name then click on "Save".



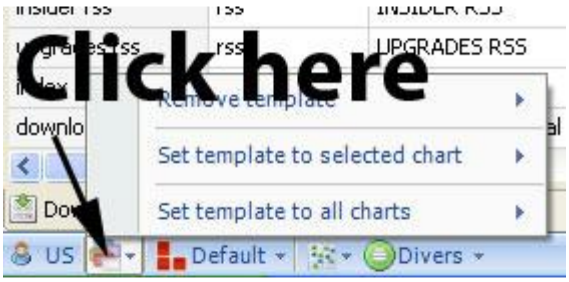
Gold

Axes & Grid | CurEPSestVS4weeksAgo

Colors	
Back color	🟨
Alpha back color	
X axis color	
Y axis color	
CandleSticks Colors	
Show single color	
Rising bar color 1	
Rising bar color 2	
Falling bar color 1	Red
Falling bar color 2	Pink
Show Rising/Fal...	☑
Rising bar bord...	Black
Falling bar bord...	Black
Stick color	Black

1.3. Remove a template

Click on the button in the toolbar as shown in the picture, select 'Remove template', and then choose the template you want to remove.

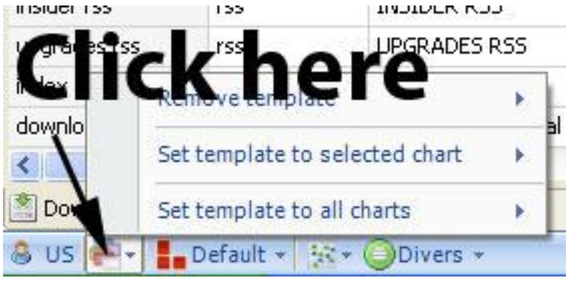


1.4. Update a pane template

On a chart's pane, right click then select 'Change template'; choose a template you want to apply for the selected pane.

1.5. Update several panes' template

Click on the button in the toolbar as shown in the picture.



To update the panes' template of the selected chart, select 'Set template to selected chart' then choose a template. To update the panes' template of all the charts, select 'Set template to all charts' then choose a template.

[TOP]

1.2.1.6 Workspaces

1. Workspaces

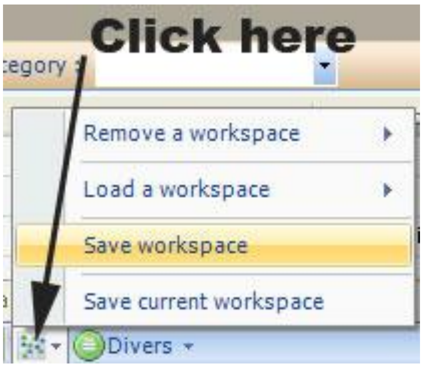
1.1. Workspaces

A workspace contains the location of forms and charts.
You can create as many workspaces as you want.

Example: You can create a workspace that contains only charts, another one that contains the download forms...

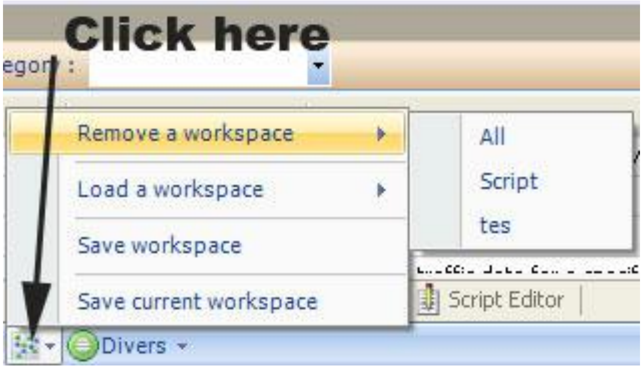
1.2. Create a workspace

To create a new workspace, click on the button, in the toolbar, as shown in the picture, select 'Save workspace' and then type a name for your newly created workspace and finally click on save.



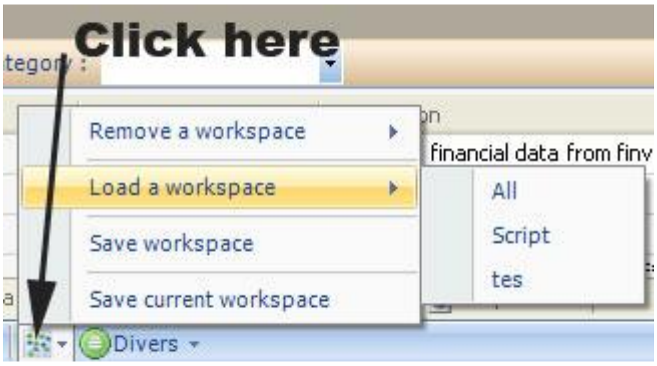
1.3. Remove a workspace

To remove a workspace, click on the button, in the toolbar, as shown in the picture, select 'Remove a workspace' and then select the workspace you would like to remove.



1.4. Load a workspace

To load a workspace, click on the button, in the toolbar, as shown in the picture, select 'Load a workspace' and then select the workspace you would like to use.


Click here
tegory :
Remove a workspace
Load a workspace
Save workspace
Save current workspace
All
Script
tes
financial data from finv
Divers

[TOP]

1.2.1.7 Events

1. Application & Thread events

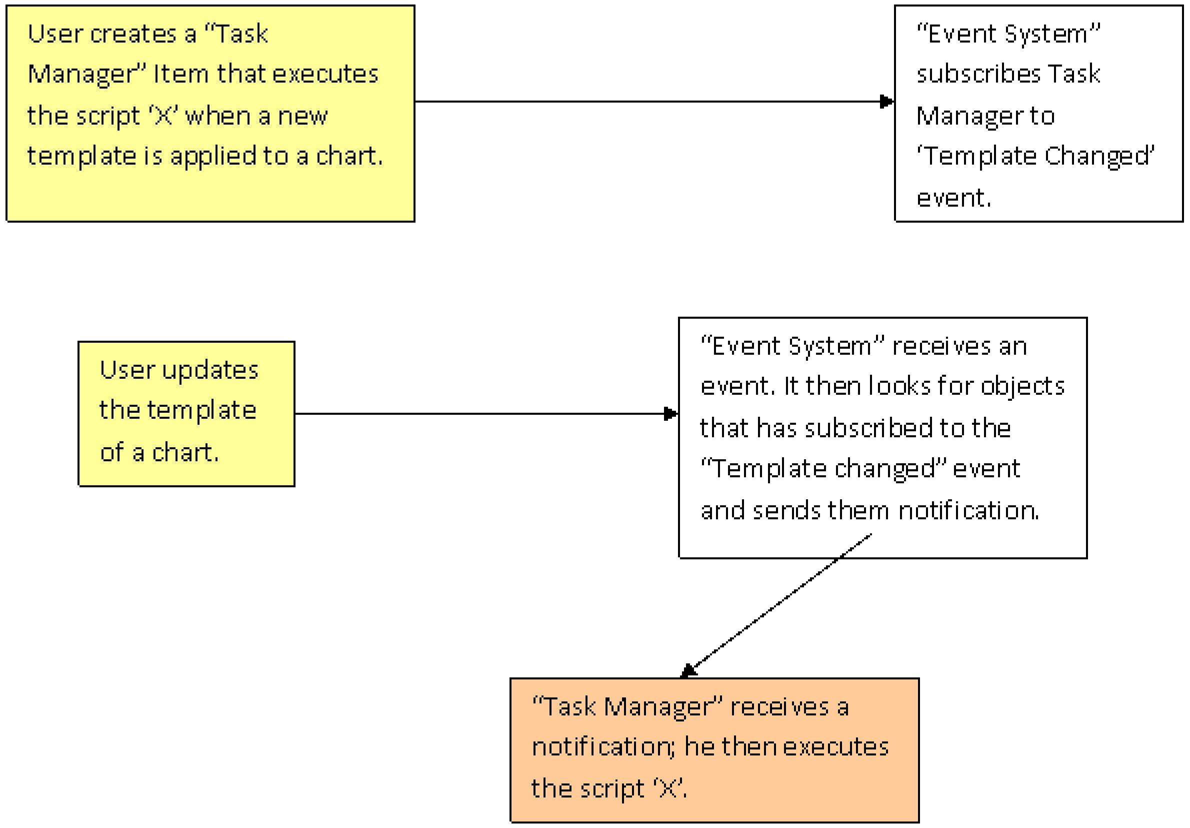
1.1. Application events

Main application and plug-ins can throw events to notify the 'Event System' that they have completed a task or that a user did an action...

Main application and plug-ins can also subscribe to the 'Event System' and receive notification about a certain event they want to catch.

The plug-in that makes the more use of these events is the 'Task Manager'. Its role is to execute scripts depending on the application or system events and on your settings.

Example of notification:



User creates a "Task Manager" Item that executes the script 'X' when a new template is applied to a chart.

"Event System" subscribes Task Manager to 'Template Changed' event.

User updates the template of a chart.

"Event System" receives an event. It then looks for objects that has subscribed to the "Template changed" event and sends them notification.

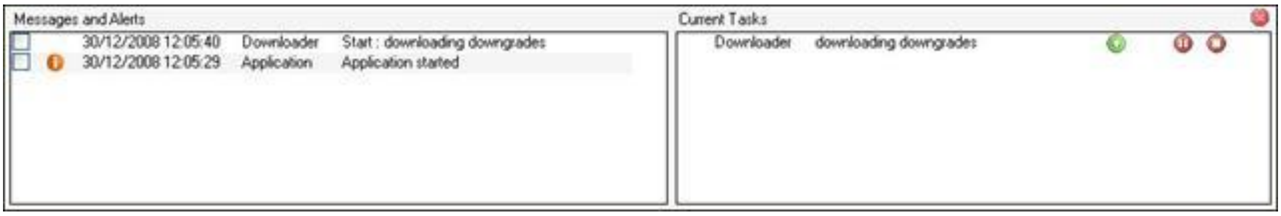
"Task Manager" receives a notification; he then executes the script 'X'.

1.2. Thread events

Thread events are events sent by plug-ins, the tasks associated with these events are generally time or CPU consuming tasks.

Example: (Downloading quotes)

The main application catches these events and adds notifications about whether these events start or end into a grid (as shown below)



```
□     30/12/2008 12:05:40   Downloader    Start : downloading downgrades
□  ●  30/12/2008 12:05:29   Application   Application started
```

```
Downloader    downloading downgrades
```

1.2.1.8 Scripting

1. Scripting

1.1. Introduction

Scripts in the QuantShare software use the .Net framework and the CSharp (c#) or Jscript.Net programming languages.

.Net Scripts extend QuantShare capabilities and make the application extremely flexible.
Scripts can be shared between QuantShare users.

There are two kinds of scripts, main script and specialized scripts.

The main script allows the user to:

- Control the application behavior
- Extract databases data
- Insert, inspect, control and modify database data
- Control the objects (composites, watch lists, list of rules...)

The specialized scripts allow the user to take control over some specific tasks.

The application offers you the possibility to write scripts for the following tasks:

Money management: Contains five events, the first and last events happen once while the other events could happen many times during the backtesting.

- OnStartSimulation: The simulator executes this script before starting the backtesting process.
- OnNewTrade: The simulator executes this script before entering a new position.
- OnCloseTrade: The simulator executes this script before exiting a position.
- OnEndPeriod: The simulator executes this script at the end of each period or bar. (In end of day back-testing, a period is a day)
- OnEndSimulation: The simulator executes this script when the back-testing process ends.

Example: Create a script that initializes an optimizable variable in the 'OnStartSimulation' event

Create a script, in the 'OnNewTrade' event, that updates the number of shares to buy depending on the trading system drawdown and the optimizable variable defined in the 'OnStartSimulation' event

Example: Create a time-series metric for your trading system using the 'OnEndPeriod' and the 'OnEndSimulation' events

Note: In money management scripts, you can define global variables that can be shared between events.

List of rules analyzer metrics: Script that let you create metrics to assess rules
Example: Calculate the average performance of a rule for each month.

Ranking system analyzer metrics: Script that let you create metrics to assess ranking systems
Example: Calculate the standard deviation of the buckets annual return.

Simulation metrics: Script that let you create metrics to assess simulations
Example: Calculate a custom metric that uses Sharpe, Sortino and some others ratios

Prediction model metrics: Script that let you create metrics to assess prediction models

Example: Calculate a custom metric that uses others already calculated metrics.

Pre-Parsing data: Script that let you manipulate text or CSV data
Example: Create a new column that is based on others columns data
Example: Update a column data so it can be parsed by the software

Post-Parsing data: Script that let you manipulate parsed data just before the application stores it

Example: Correct bad quotes

URL-Script: Script that let you dynamically create URLs for the downloader to download. The data is then passed to the Pre-Parsing and Post-Parsing scripts if they are implemented.
Example: Grab the download URL from a website that dynamically changes the URL path every day.

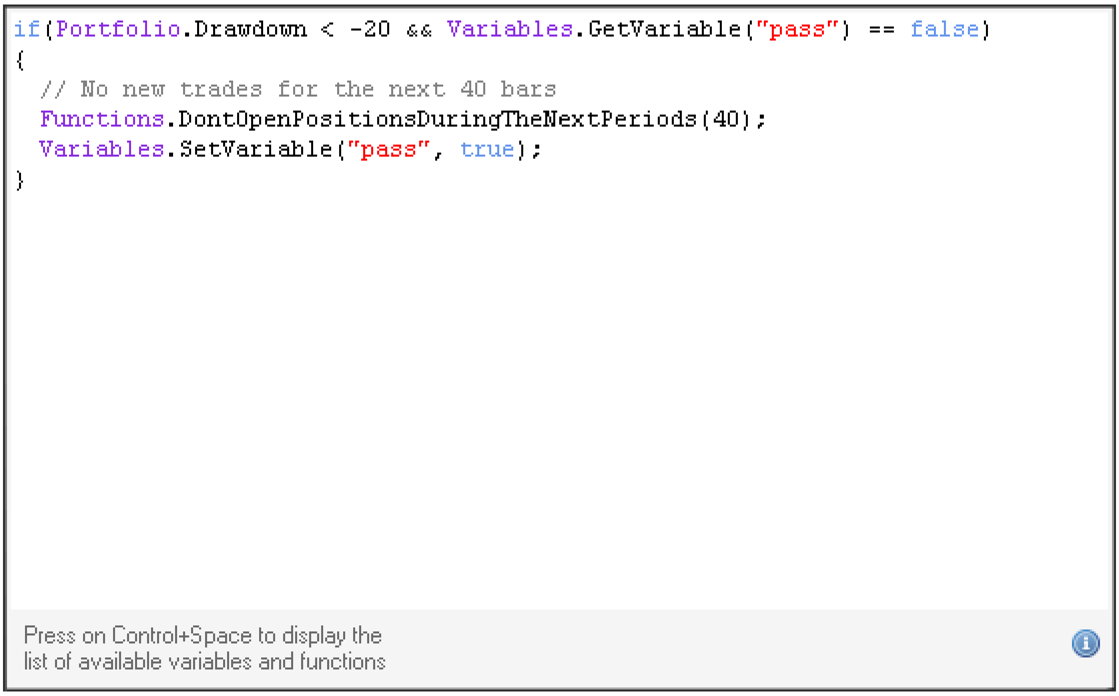
Functions: Function scripts let you create QuantShare functions. These functions could be used in charting, simulation...

Example: Correct a custom indicator that is not available in the QuantShare indicators list

1.2. Script Control

.Net script shares the same control; the control is an advanced text box control where you can type in your script.
You can press on CTRL+SPACE at any moment to display the list of functions or variable that you can use in the current

script.
Each script contains its own functions and classes.
You can click on the right icon to display information about the current script.



```
if(Portfolio.Drawdown < -20 && Variables.GetVariable("pass") == false)
{
  // No new trades for the next 40 bars
  Functions.DontOpenPositionsDuringTheNextPeriods(40);
  Variables.SetVariable("pass", true);
}
```

Press on Control+Space to display the
list of available variables and functions

1.3. Global Variables

The Global variable class lets you define global variables.
This class is present in all scripts (main script and specialized scripts).
Its main functions are: GetVariable and SetVariable.
Example: Global.SetVariable("myvar", 20);
The application will now create a global variable "myvar" that has a value of 20.
You can access this variable from any script. This will allow scripts to interact with each other.

1.4. Create functions inside a script:

You can create inside a script as many functions as you want. You just need to tell the application where the main code
ends. This is done, by using the following keyword: #functions#
Example:

```
double a = 0;
for(int i=0;i<10;i++)
{
    a = Sum(a, 1);
}

MessageBox.Show(a.ToString());

#functions#
public double Sum(double a, double b)
{
    return a + b;
}
```

This C# script calls the function "Sum" ten times.

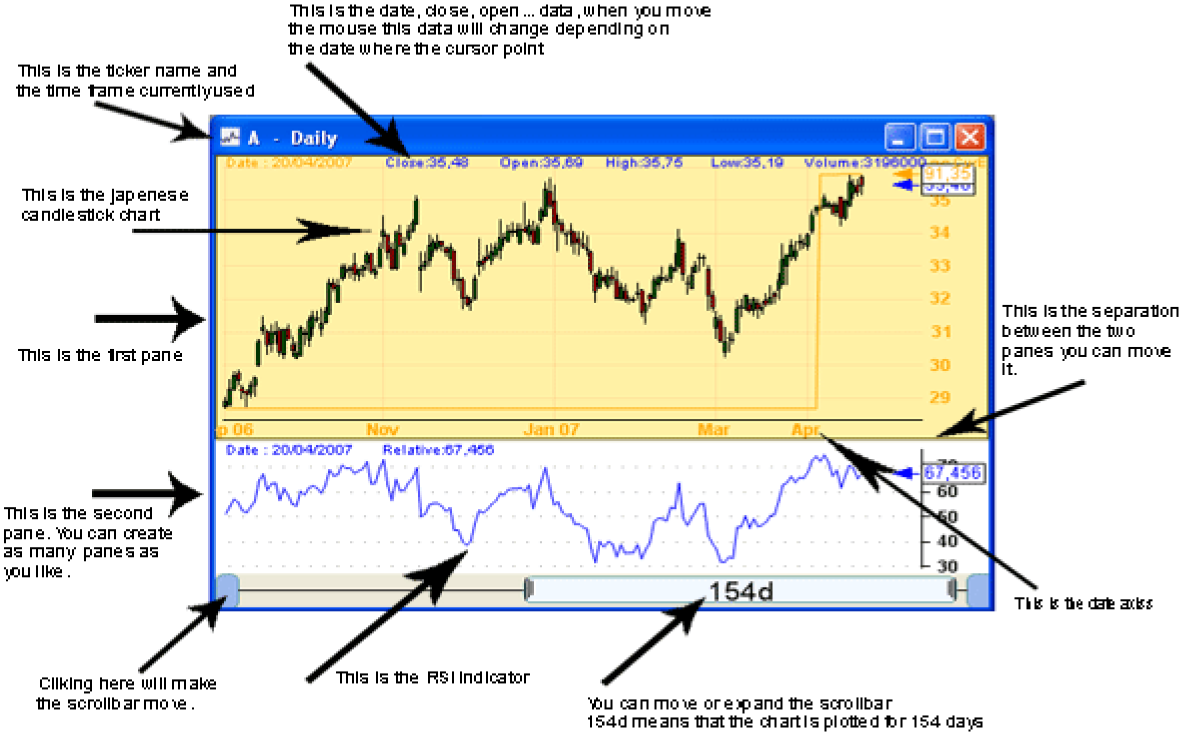
[TOP]

1.2.2 Charting

1.2.2.1 Charts

1. Charts

1.1. Introduction


This is the date, close, open ... data , when you move the mouse this data will change depending on the date where the cursor point
This is the ticker name and the time frame currently used
A - Daily
Date : 20/04/2007 Close:35,48 Open:35,69 High:35,75 Low:35,19 Volume:3196000
91,35
35
34
33
32
31
30
29
p 06
Nov
Jan 07
Mar
Apr
This is the japenese candlestick chart
This is the first pane
This is the separation between the two panes you can move it.
Date : 20/04/2007 Relative:67,456
67,456
60
50
40
30
This is the second pane. You can create as many panes as you like .
This is the date axis
154d
Cliking here will make the scrollbar move .
This is the RSI indicator
You can move or expand the scrollbar 154d means that the chart is plotted for 154 days

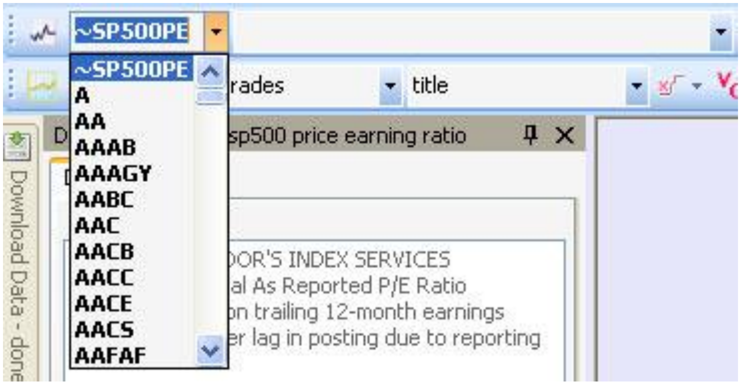
1.2. Basic operations

1.2.1. Open a chart

View->New chart

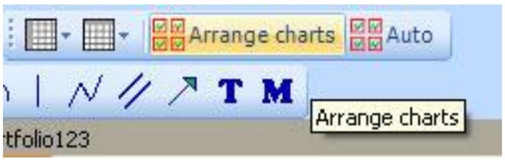
1.2.2. Update chart symbol

Select a chart, and then select a symbol from the tool bar; the chart symbol will be automatically updated.

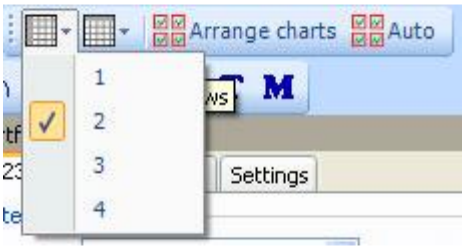


1.2.3. Arrange charts

To arrange charts, click on the button shown below.



You can also specify how to arrange charts by specifying the number of rows and columns.



1.2.4. Create linked chart

A linked chart is a chart that is linked with another one.
If chart 'A' is linked to chart 'B', then if we change the chart 'B' symbol then the symbol of chart 'A' will be updated as well.

To create a linked chart, select a chart then select (*View->New linked chart)*

1.2.5. Shortcuts

To display a list of shortcuts, click on 'Divers' in the bottom tool bar, then click on 'List of Shortcuts'.

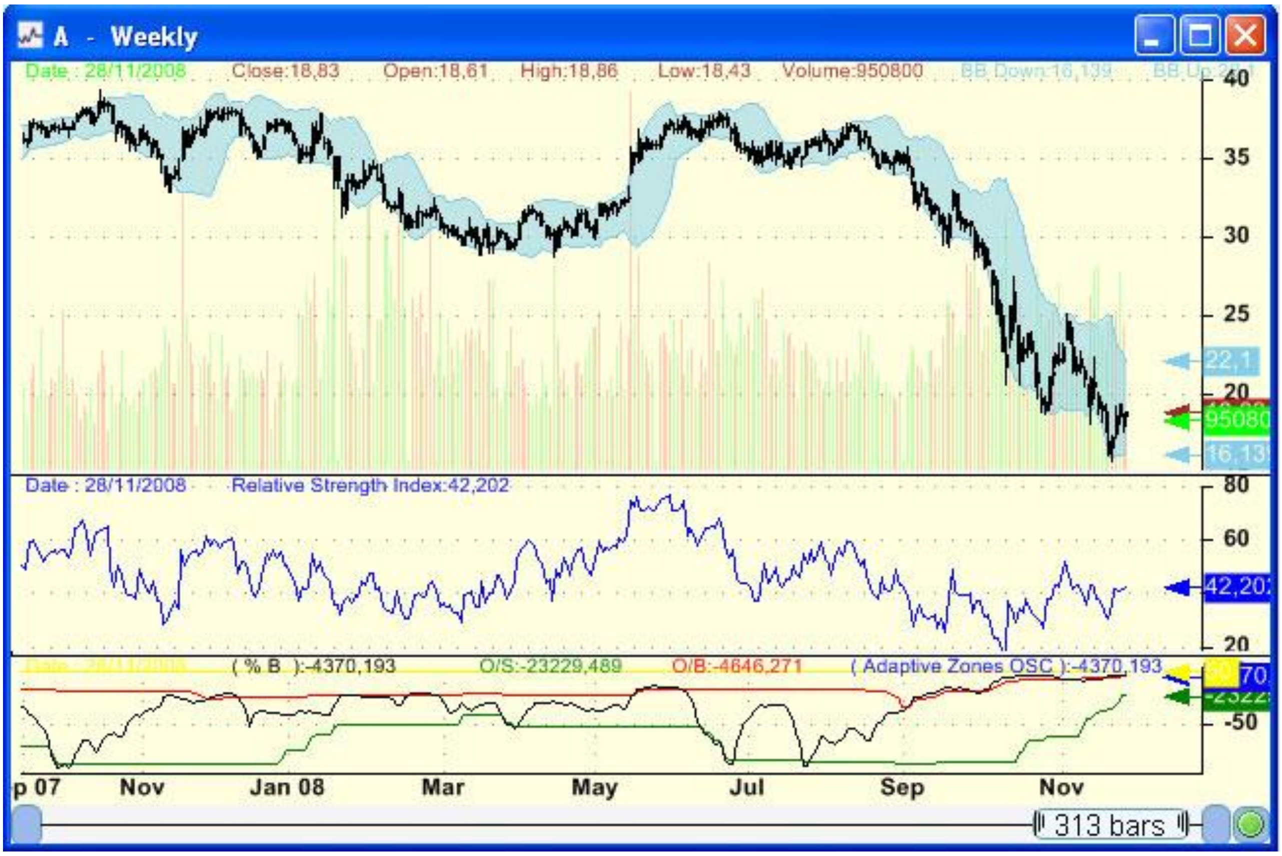
1.2.6. Zooming

To zoom into a chart, click on CTRL then use the mouse wheel to zoom in and zoom out.

1.3. Panes

Each chart is a collection of panes; a chart can contain one or several panes.
Each pane contains a template and one or many formula files.

Example of chart with three panes:

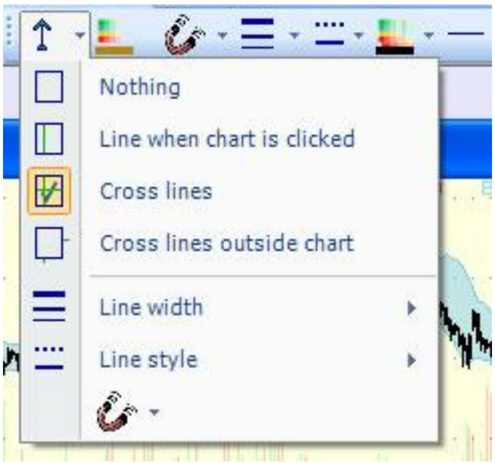


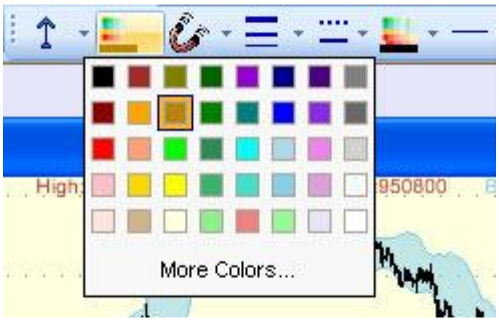
1.4. Select Line

There are three select tools available:

- Line when chart clicked
- Cross lines
- Cross lines outside chart

You can specify some settings like color, line width, Stick to (See Drawing tools)...





1.5. Time-Frames

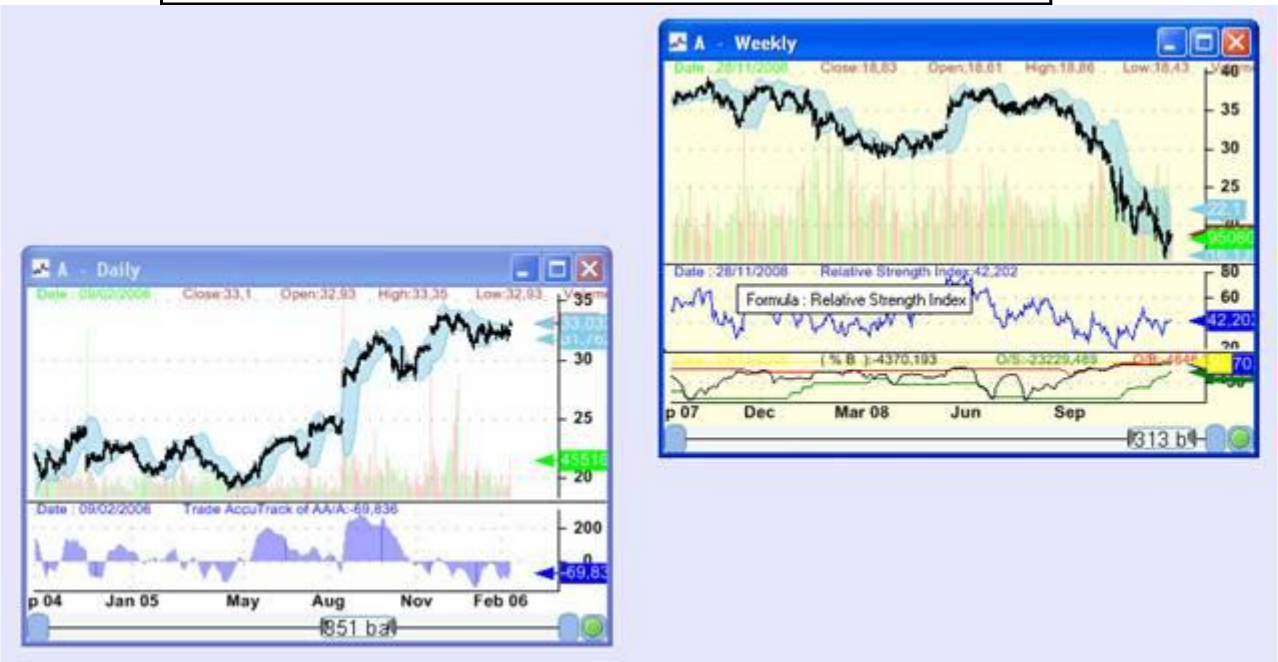
Available periods are day, week, month, year, and custom.
Custom periods let you select the number of days to use as timeframe.

To change a chart timeframe, you have to right click on a chart, select 'Chart Timeframe' in the menu and then select the timeframe.

1.6. Drag & Drop

You can drag and drop formulas from a pane to another one.
If you click on a curve or a bar, the application will detect which formula this curve or this bar belongs to, you just have then to drag and drop this formula to another pane.

Click on a study then drag & drop to another pane



You can also drag and drop a formula to the widget or bookmark panel.
Note that unlike the majority of the other items (Trading systems, watch-lists, composites ...), when the drag & drop process completes, a copy is saved and not a reference.
This means that if you drag and drop a chart formula to the bookmark panel, then you update the formula in that panel, the chart formula will not change.

1.7. Plot Formulas

Charts are formula based.
You can create charts without writing any line of code, or you can create complex charts using the QuantShare Language.

You can plot candlesticks, lines, bars, smoothed lines, step lines, symbols, and texts.

There is also a special type of graphs called fixed graphs.

You can plot for example something like "Average daily performance for each day of the week", or "number of days up and days down for each month"...

Here is the formula syntax:

PLOTFIXED(vector1, vector2, string vector, calculation function, description, color, ChartStyle, Style);

Let us imagine we want to calculate the average closing performance for each day of week.

Vector1: (Day of week) (one means Monday, two for Tuesday...)

1	2	3	4	5	1	2	3	4	5

Vector2: (Close price)

10	11	12	10	8	6	9	11	13	12

Intermediate vectors:

X vector:

1	2	3	4	5

Y vector:

10	11	12	10	8
6	9	11	13	12

Then if we have selected "Sum" as calculation function, we will have:

Y vector

16 (10+6)	20	23	23	20

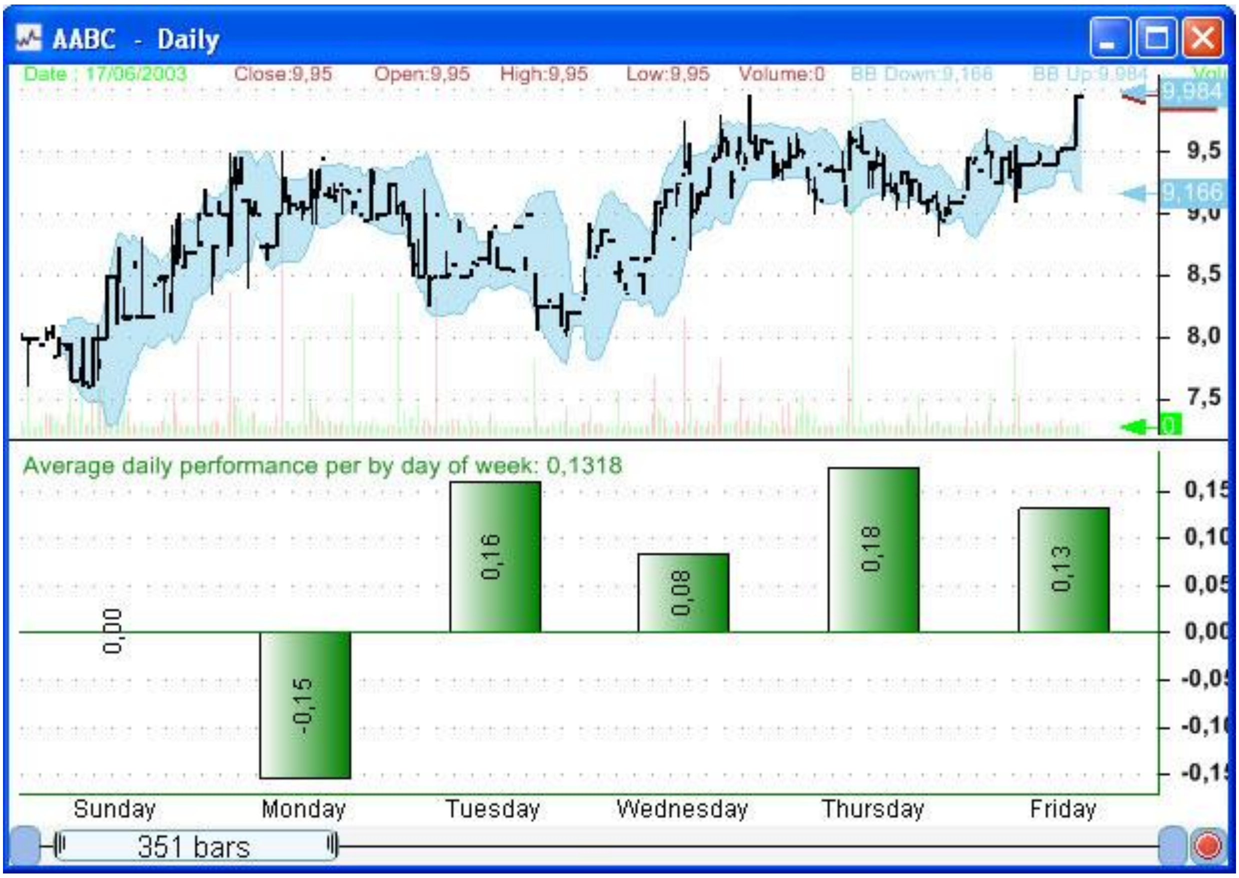
In our case, we want the average closing price so Y vector will look like this:

8 (10+6)/2	10	11.5	11.5	10

At the end, we plot the graph with X vector as X-axis values and Y vector as Y-axis values.

Example formula:

```
// Average daily performance per by day of week
// One day price change
performance = perf(close, 1);


// Day of week
dayofw = dayofweek();


// Change day of week numeric values to text format
text = StringReplace(dayofw ,0,"Sunday");
text = StringReplace(text,1,"Monday");
text = StringReplace(text,2,"Tuesday");
text = StringReplace(text,3,"Wednesday");
text = StringReplace(text,4,"Thursday");
text = StringReplace(text,5,"Friday");
// Plot fixed chart
PLOTFIXED(dayofw, performance,text , iAvg,"Average daily performance per by day of
week",colorGreen,ChartBar,StyleSymbolDiamond);
```



1.7. Formulas Files

Charts don't contain reference to formula files but instead contain formula contents.
This means that even if you remove or update a formula file that a chart was build upon, the chart won't change.

Right click on a chart, and then click on 'Edit formula' to edit a chart formula.
Only the current chart will be affected if you update the formula or change its name.

You can save the content of this formula to a file and use it later with other charts.

[TOP]

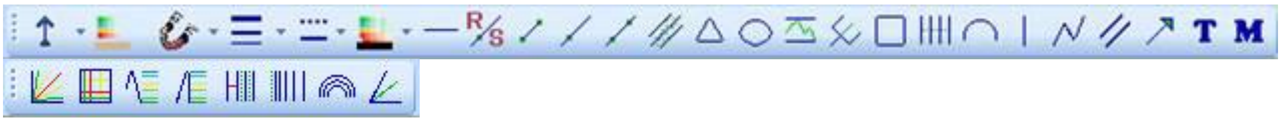
1.2.2.2 Drawing tools

1. Drawing tools

1.1. Drawing tools list

List of the available drawing tools:

- Line
- Extended line
- Horizontal line
- Vertical line
- Regression channel
- Triangle
- Ellipse
- Trend line
- Pitchfork
- Rectangle
- Cycle line
- Arc
- Zigzag
- Parallel lines
- Arrow
- Text
- Marker
- Gann fan
- Gann square
- Fibonacci extension
- Fibonacci retracement
- Fibonacci time extension
- Fibonacci time zones
- Fibonacci arc
- Fibonacci fan



1.2. Usage of the drawing tools

Select a drawing item then click on the chart, point the mouse to a second location then click on the left mouse button.

Some drawing tools require three points or more.

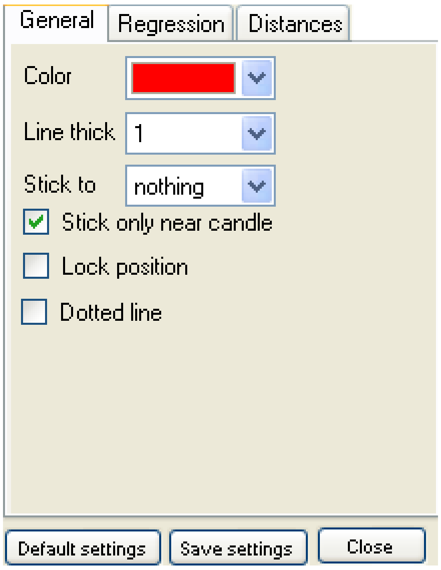
1.3. Drawing tools settings

Each drawing tool has its own settings. You can update these settings by clicking on the right mouse button when the cursor is pointing on the drawing item.

Some settings are common to all the drawing tools, while others are specific.

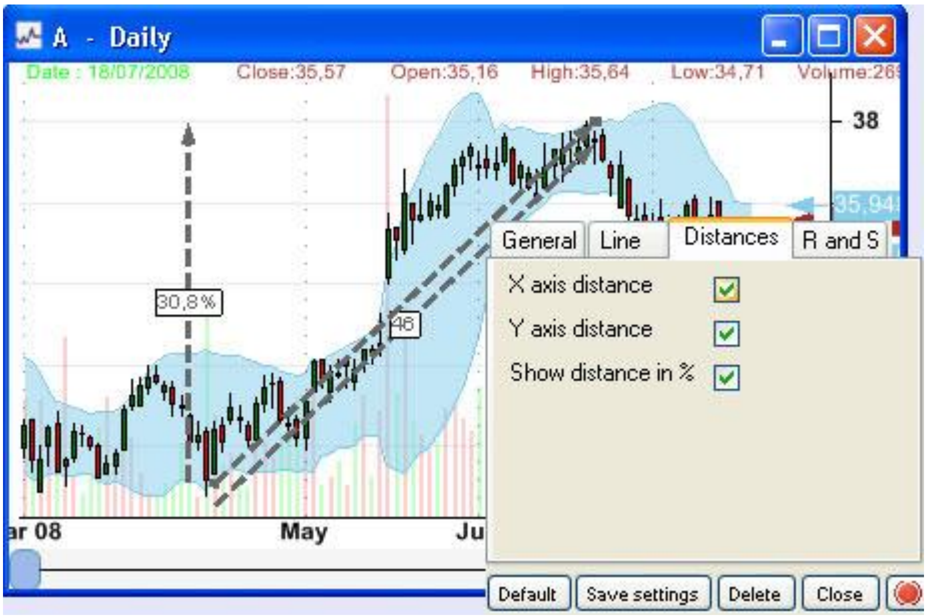
The common settings are:

Line color
Line width
Line dotted or not
Locked or not
Stick to (high, low, close, open, all and no stick)
Stick only near candle



Color 🟥
Line thick 1
Stick to nothing
☑ Stick only near candle
□ Lock position
□ Dotted line

Examples of specific settings:
Line: In the line tool, a setting named 'X axis distance' and another one named 'Y axis distance' let you draw distances on a chart.

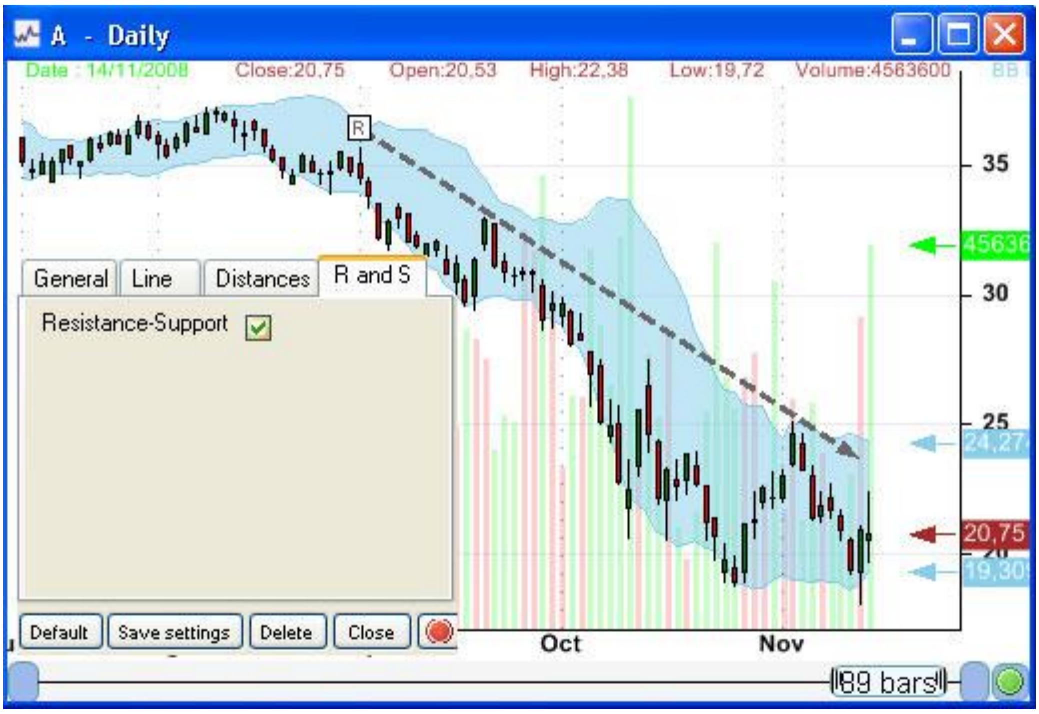


X axis distance ☑

Y axis distance ☑

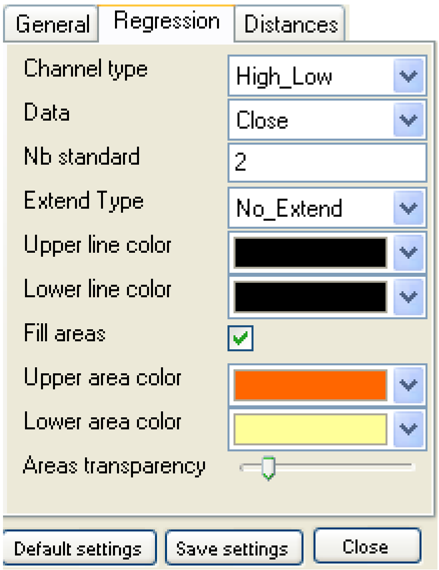
Show distance in % ☑

The line can be transformed to a 'Support resistance line'.
On a support resistance line, the letter 'S' means that your line now is a support, while a letter 'R' means that the line is a resistance.

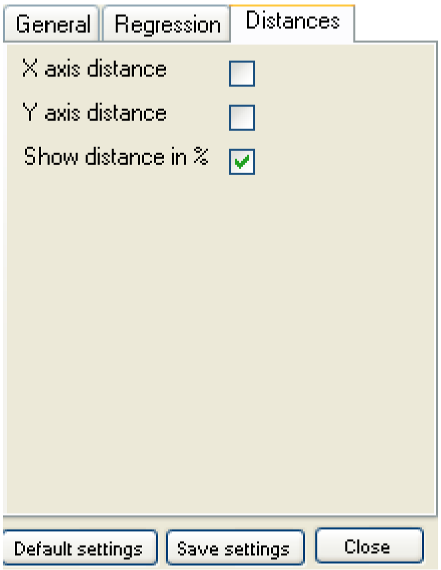


Resistance-Support ☑

Regression channel:



Channel type: High_Low
Data: Close
Nb standard: 2
Extend Type: No_Extend
Upper line color: ⬛
Lower line color: ⬛
Fill areas: ☑
Upper area color: 🟧
Lower area color: 🟨
Areas transparency



X axis distance: □
Y axis distance: □
Show distance in %: ☑

1.4. Remove a drawing tool

Right click on a drawing tool to open the settings form; click on 'Delete'.
To delete all the drawing tools, right click on the chart then select 'Delete all studies' in the menu.

1.5. Saving drawing tools

To save a drawing tool settings:

- Right click on that drawing item
- Update the settings according to your preference
- Click on 'Save Settings' button

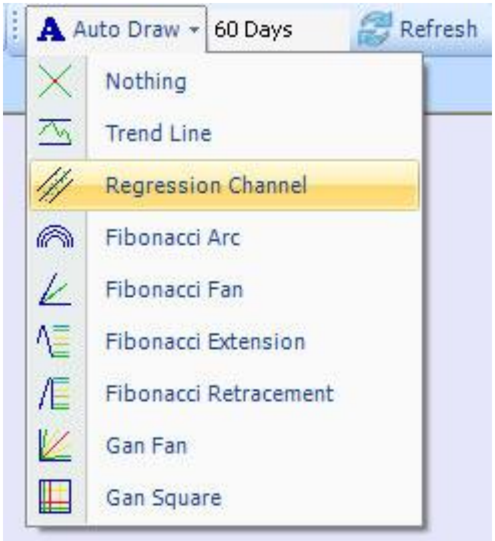
[TOP]

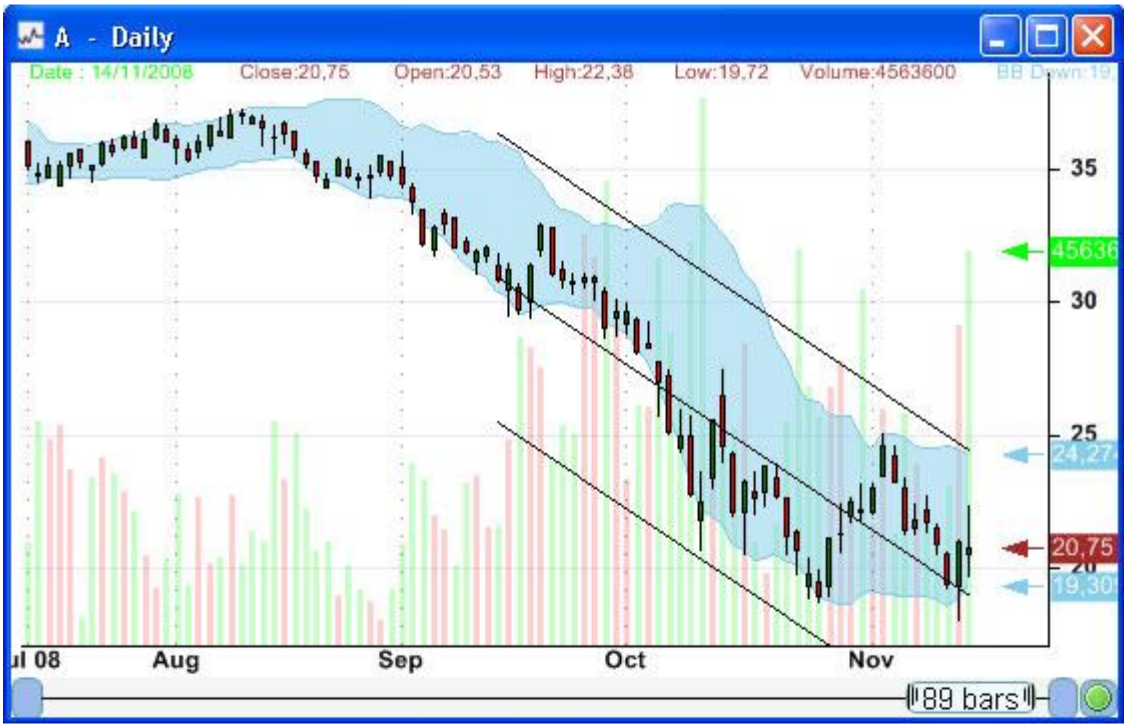
1.2.2.3 Auto drawing tools

1. Auto drawing tools

Auto drawing is a feature that automatically draws on a chart whenever a chart symbol changes or a new chart window opens.

You have to choose which drawing tools you want to use, then specify the number of days to use and the application will take care of the drawing.
The drawing generated by this feature is locked.
To unlock it, just right click on the drawing then uncheck 'lock position'.





[TOP]

1.2.3 Symbols

1.2.3.1 Symbols

1. Symbols

A symbol references a stock, an option, an index, a composite...

1.1. Add Symbols

To add symbols to your database (*Symbol->Add Symbols*)
You can add symbols in two ways:

1 – Insert comma separated symbols in the field then click on 'Add'

2 – Specify a file that contains a list of comma separated symbols.

You can also add symbols by downloading a list of symbols from the Sharing Server.
List of symbols are a collection of symbols shared by other users.
After you download a list of symbols. Select "Symbols" in the application main menu, then click on "Add-Remove from list of symbols". Click on "Select" to select a list, then click on "Add".

1.2. Remove Symbol

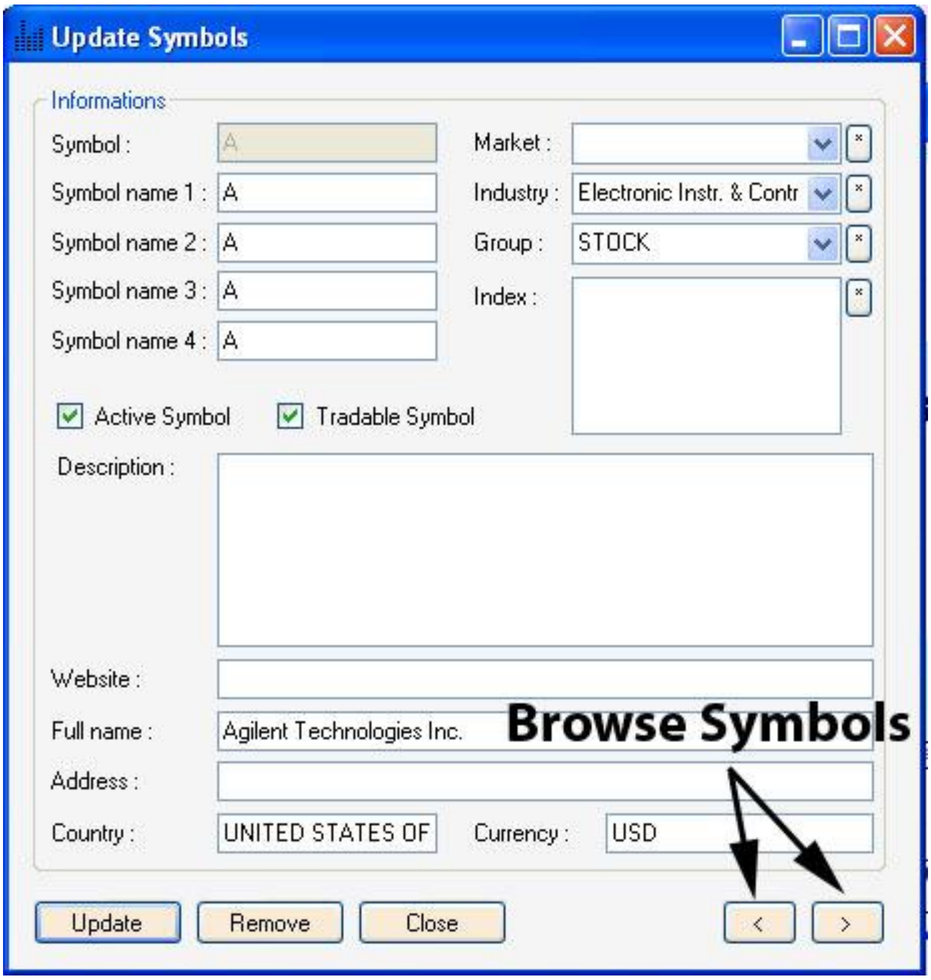
Select (*Symbol -> Update Symbols*)
Browse through the symbols and click on 'Remove' on the symbol you want to delete.

NB: Use the bulk symbol updater to remove many symbols at once.

You can also remove symbols that belong to a list of symbols. To do so, select "Symbols" in the application main menu, then click on "Add-Remove from list of symbols". Click on "Select" to select a list, then click on "Remove".

1.3. Update Symbols

Select (*Symbol->Update Symbols*)



Update Symbols

Informations

Symbol :	A	Market :	
Symbol name 1 :	A	Industry :	Electronic Instr. & Contr
Symbol name 2 :	A	Group :	STOCK
Symbol name 3 :	A	Index :	
Symbol name 4 :	A		

☑ Active Symbol ☑ Tradable Symbol

Description :

Website :

Full name : Agilent Technologies Inc.

Address :

Country : UNITED STATES OF Currency : USD

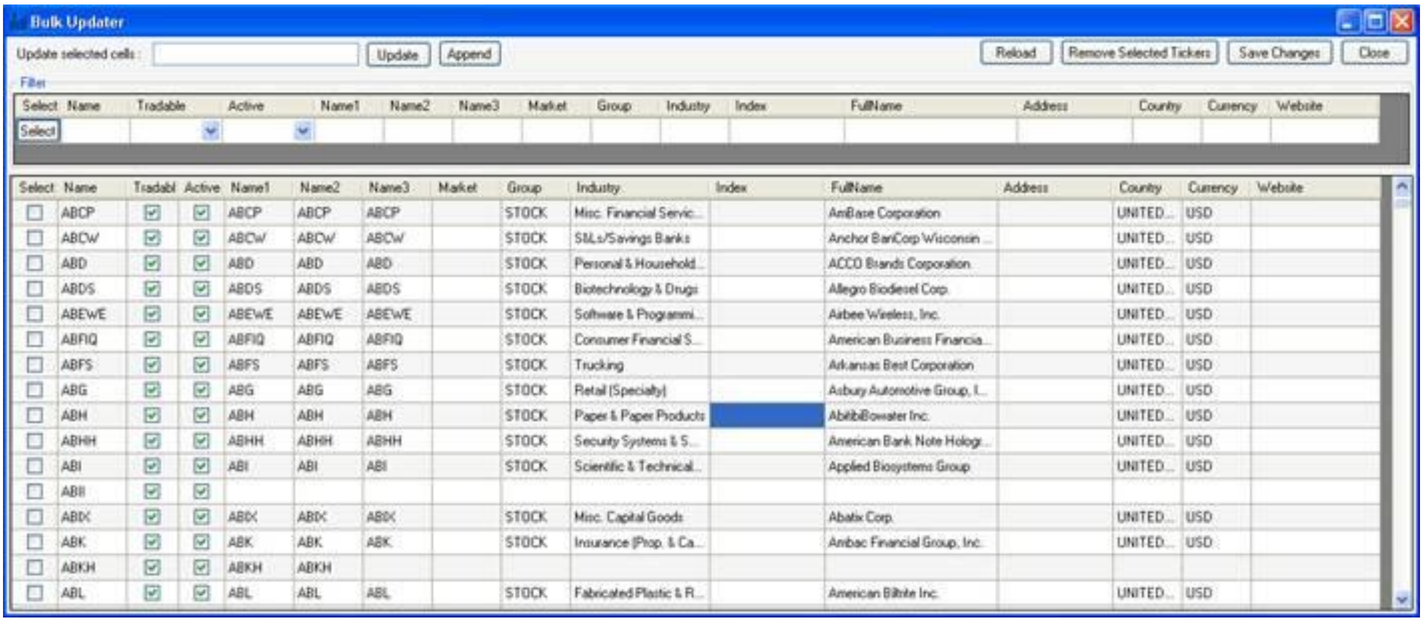
Browse Symbols

You can update all symbols' information within this form.
After you have done, click on 'Update'.

NB: Use the bulk symbol updater to update many symbols at once.

1.4. Bulk symbol updater

The 'Bulk Symbol' is used to update many symbols at once. (*Symbol -> Bulk Updater*)



Bulk Updater

Update selected cells :

Select	Name	Tradable	Active	Name1	Name2	Name3	Market	Group	Industry	Index	FullName	Address	Country	Currency	Website

Select	Name	Tradabl	Active	Name1	Name2	Name3	Market	Group	Industry	Index	FullName	Address	Country	Currency	Website
□	ABCP	☑	☑	ABCP	ABCP	ABCP		STOCK	Misc. Financial Servic...		AmBase Corporation		UNITED...	USD	
□	ABCW	☑	☑	ABCW	ABCW	ABCW		STOCK	S&Ls/Savings Banks		Anchor BanCorp Wisconsin ...		UNITED...	USD	
□	ABD	☑	☑	ABD	ABD	ABD		STOCK	Personal & Household...		ACCO Brands Corporation		UNITED...	USD	
□	ABDS	☑	☑	ABDS	ABDS	ABDS		STOCK	Biotechnology & Drugs		Allegro Biodiesel Corp.		UNITED...	USD	
□	ABEWE	☑	☑	ABEWE	ABEWE	ABEWE		STOCK	Software & Programmi...		Airbee Wireless, Inc.		UNITED...	USD	
□	ABFIQ	☑	☑	ABFIQ	ABFIQ	ABFIQ		STOCK	Consumer Financial S...		American Business Financia...		UNITED...	USD	
□	ABFS	☑	☑	ABFS	ABFS	ABFS		STOCK	Trucking		Arkansas Best Corporation		UNITED...	USD	
□	ABG	☑	☑	ABG	ABG	ABG		STOCK	Retail (Specialty)		Asbury Automotive Group, I...		UNITED...	USD	
□	ABH	☑	☑	ABH	ABH	ABH		STOCK	Paper & Paper Products		AbitibiBowater Inc.		UNITED...	USD	
□	ABHH	☑	☑	ABHH	ABHH	ABHH		STOCK	Security Systems & S...		American Bank Note Hologr...		UNITED...	USD	
□	ABI	☑	☑	ABI	ABI	ABI		STOCK	Scientific & Technical...		Applied Biosystems Group		UNITED...	USD	
□	ABII	☑	☑												
□	ABIX	☑	☑	ABIX	ABIX	ABIX		STOCK	Misc. Capital Goods		Abaxis Corp.		UNITED...	USD	
□	ABK	☑	☑	ABK	ABK	ABK		STOCK	Insurance (Prop. & Ca...		Ambac Financial Group, Inc.		UNITED...	USD	
□	ABKH	☑	☑	ABKH	ABKH										
□	ABL	☑	☑	ABL	ABL	ABL		STOCK	Fabricated Plastic & R...		American Biltrite Inc.		UNITED...	USD	

1.5. Create a list of symbols

There are two ways to create a list of symbols:

- Select "Symbols" then click on "Create a list of symbols". Specify the symbols to include using the "Symbols Selection Control" then click on "Create"

- Select "Symbols" then click "Create a list of symbols from a file". Type a separator then click on "Load file" to load a file that contains your symbols. The first row lets you specify the type of each column. For example: the symbol name should be associated with the keyword "name". After you finish, click on "Create List".

You can use this list to add symbols in another account, or you can share this list, so other members can use it. They will thank you.

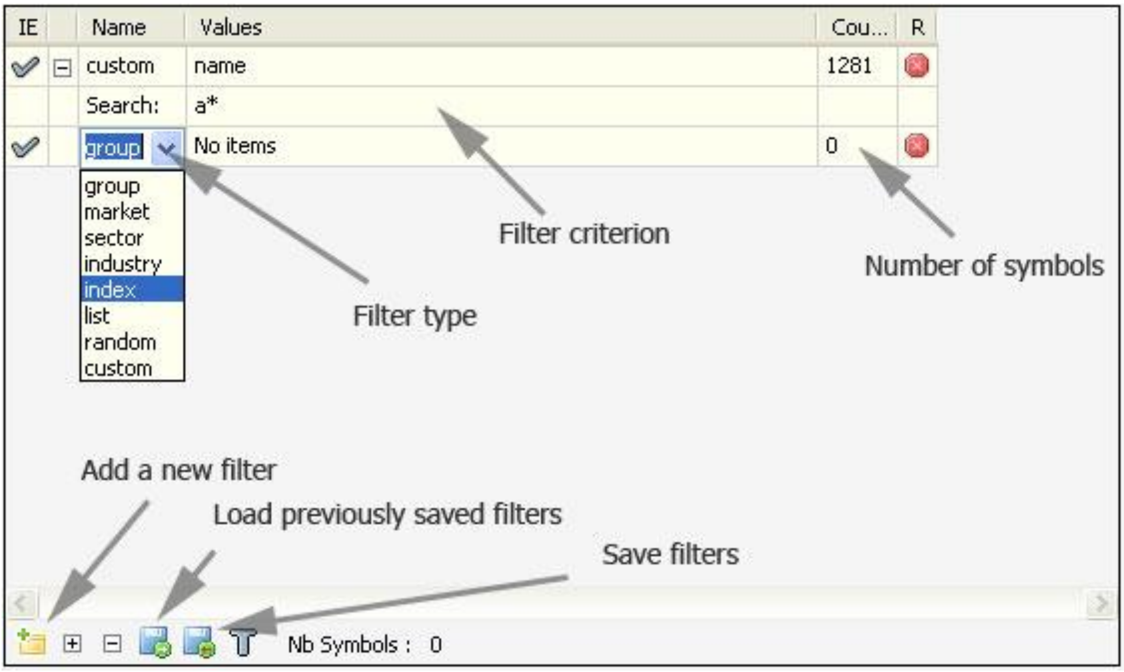
[TOP]

1.2.3.2 Symbols Selection

1. Symbols Selection

1.1. Symbols selection control

You will often need to select certain symbols among the list of all available symbols.
Whether it is for creating a composite, a trading system or a watch-list, the process of symbols selection is very easy. We have created a unique control that is used by all the plug-ins.



IE		Name	Values	Cou...	R
✓	⊟	custom	name	1281	●
		Search:	a*		
✓		group	No items	0	●

Nb Symbols : 0

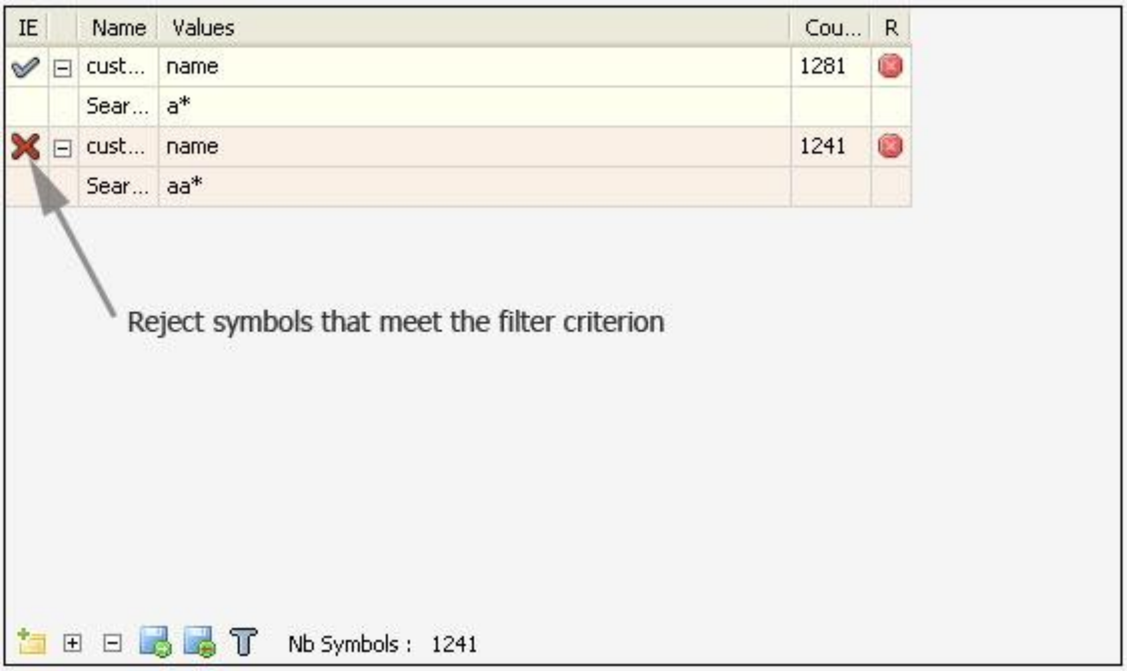
You can select symbols using a single or a combination of filters:

- By group: select symbols that belongs to specified groups

- By sector: select symbols that belongs to specified sectors

- By Industry: select symbols that belongs to specified industries
- By Index: select symbols that belongs to specified indexes
- By Market: select symbols that belongs to specified markets
- By Entering symbols (list): click on 'select symbols', then enter manually the symbols you want to include.
- Randomly: type a value from 0 to 100. Zero means select no symbols, and 100 means select all symbols.
- Custom filter: create a custom filter (see custom filter section)

In order for a symbol to be selected, it must pass all the filters.
You can also make the filter act differently by clicking on the icon as shown in the picture below.



IE		Name	Values	Cou...	R
✓	⊟	cust...	name	1281	●
		Sear...	a*		
✗	⊟	cust...	name	1241	●
		Sear...	aa*		

Reject symbols that meet the filter criterion

Nb Symbols :   1241

Example:

Total symbols: 8000.

Filter1: Random; 50 percent. Number of symbols that passed the filter: 4000.

Number of symbols selected: 4000 symbols.

Filter2: Symbols whose names begin with 'a'. Number of symbols that passed the filter: 700.

Number of symbols selected: 380 symbols.

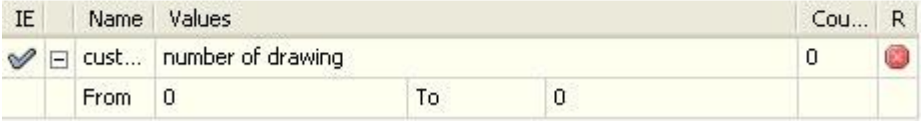
Because of the way the 'Random' filter is used in this selection process, if you run these filters again, you may end up with slightly different results.

1.2. Custom Filter

When selecting a custom filter, you can use pre-calculated data to filter symbols.
A Pre-calculated data can be a number, a text, a date or a boolean (true or false).
For each type you have to enter or select one or more values in order to create the filter.

Numeric:

Example: number of drawing objects.

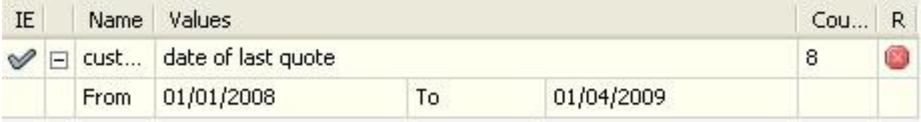


IE		Name	Values			Cou...	R
✓	⊟	cust...	number of drawing			0	☒
		From	0	To	0		

You have to specify the start and the end values. All symbols that have a number of drawing objects within these values will be selected.

Date:

Example: date of last quote

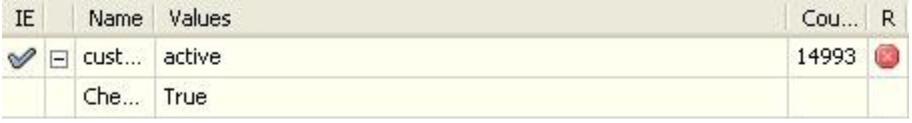


IE		Name	Values			Cou...	R
✓	⊟	cust...	date of last quote			8	☒
		From	01/01/2008	To	01/04/2009		

You have to specify the start date and the end date.

Boolean:

Example: active symbols (active)

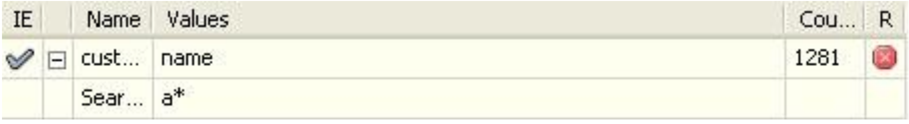


IE		Name	Values	Cou...	R
✓	⊟	cust...	active	14993	☒
		Che...	True		

You have to choose between TRUE or FALSE.

Text:

Example: symbol name (name)



IE		Name	Values	Cou...	R
✓	⊟	cust...	name	1281	⊗
		Sear...	a*		

You have to enter one or more values. Values are separated by semicolon.

A value can contain the following characters: (multiply *), (point .)

Examples:

A: select the symbol which name is 'A'.

A*: select symbols whose name start with 'A'.

A.(1 point): select symbols whose names start with 'A' and have only two characters.

A..(2 points): select symbols whose names start with 'A' and have only three characters.

A: select symbols whose names contain the character 'A'.

A*;B*: select symbols whose names start with 'A' or 'B'.

1.2.4 Data

1.2.4.1 Databases

1. Databases

There are two types of databases:

- Quote database
- Custom database

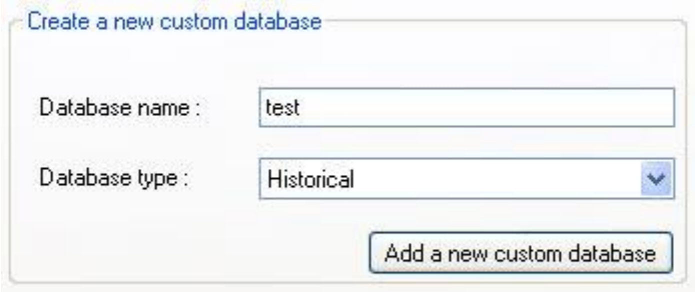
The quote database contains symbols quotes, while the custom database can contain any text or numeric data.
Example: news database, fundamental database

The custom database can be a daily or an intraday database.
A daily database is a database where each row represents a day while in an intraday database each row represents a date and a time.

1.1. Create a custom database

In 'Data' menu; click on 'Edit databases'. (*Data->Edit databases*)
In the new form, click on the tab' name 'Creator'.

Type in the name and the type (Historical or Intraday) of the database then click on 'Add new custom database'.

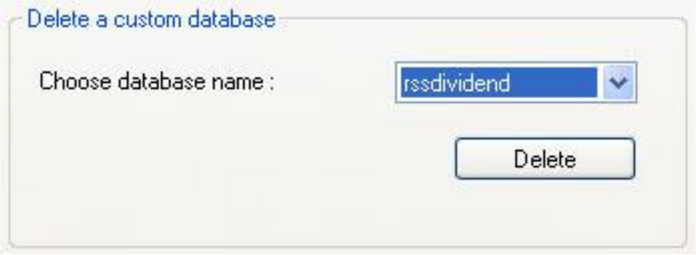


Database name : test

Database type : Historical

1.2. Delete a custom database

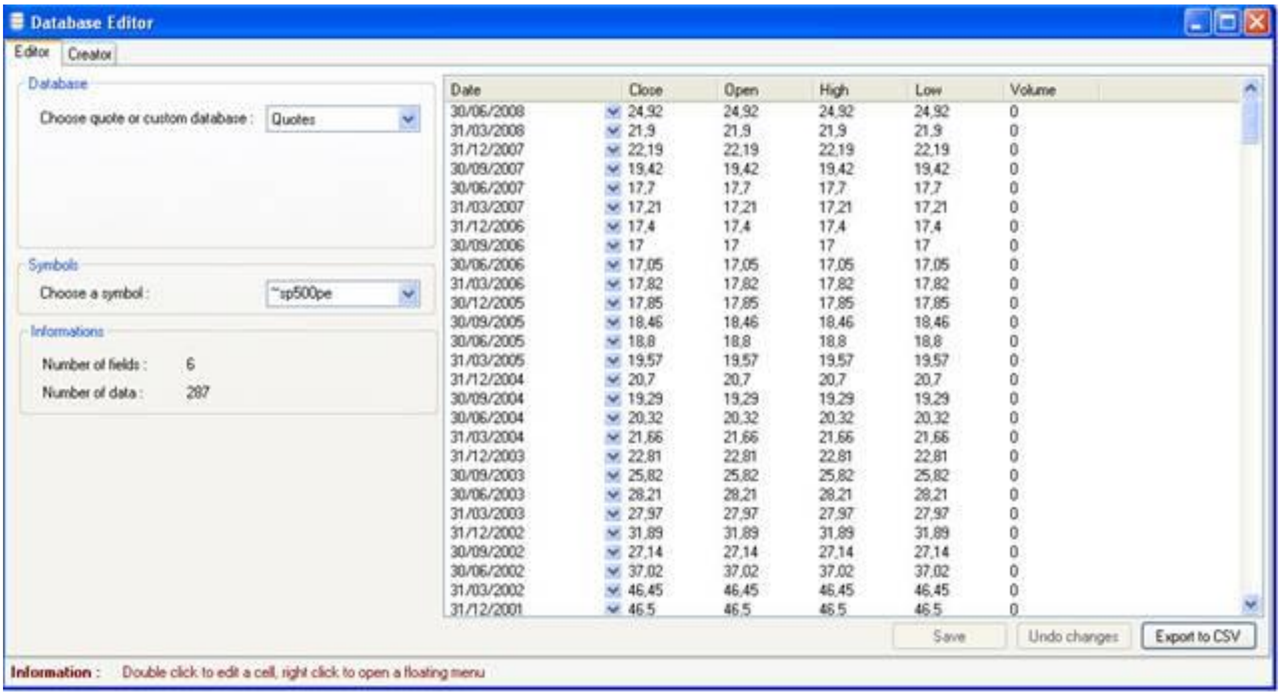
In 'Data' menu; click on 'Edit databases'. (*Data->Edit databases*)
In the new form, click on the tab' name 'Creator'.
Select a database in the 'Choose database name' combo box, then click on 'Delete'.



Choose database name : rssdividend

1.3. View data contained in a custom database

In 'Data' menu; click on 'Edit databases'.
In the new form, select 'Custom' in the 'Choose quote or custom database' combo box, then select a database name in the 'Choose database name' combo box.
Browse symbols by selecting a symbol in the 'Choose a symbol'. The database data, if any, should appears on the grid.



Choose quote or custom database : Quotes

Choose a symbol : ~sp500pe

Number of fields : 6
Number of data : 287

Date	Close	Open	High	Low	Volume
30/06/2008	24,92	24,92	24,92	24,92	0
31/03/2008	21,9	21,9	21,9	21,9	0
31/12/2007	22,19	22,19	22,19	22,19	0
30/09/2007	19,42	19,42	19,42	19,42	0
30/06/2007	17,7	17,7	17,7	17,7	0
31/03/2007	17,21	17,21	17,21	17,21	0
31/12/2006	17,4	17,4	17,4	17,4	0
30/09/2006	17	17	17	17	0
30/06/2006	17,05	17,05	17,05	17,05	0
31/03/2006	17,82	17,82	17,82	17,82	0
30/12/2005	17,85	17,85	17,85	17,85	0
30/09/2005	18,46	18,46	18,46	18,46	0
30/06/2005	18,8	18,8	18,8	18,8	0
31/03/2005	19,57	19,57	19,57	19,57	0
31/12/2004	20,7	20,7	20,7	20,7	0
30/09/2004	19,29	19,29	19,29	19,29	0
30/06/2004	20,32	20,32	20,32	20,32	0
31/03/2004	21,66	21,66	21,66	21,66	0
31/12/2003	22,81	22,81	22,81	22,81	0
30/09/2003	25,82	25,82	25,82	25,82	0
30/06/2003	28,21	28,21	28,21	28,21	0
31/03/2003	27,97	27,97	27,97	27,97	0
31/12/2002	31,89	31,89	31,89	31,89	0
30/09/2002	27,14	27,14	27,14	27,14	0
30/06/2002	37,02	37,02	37,02	37,02	0
31/03/2002	46,45	46,45	46,45	46,45	0
31/12/2001	46,5	46,5	46,5	46,5	0

Information : Double click to edit a cell, right click to open a floating menu

1.4. Add a field in your database

Select a custom database.
Type the field name in the text box near the button 'Add new field', select the field type in the combo box located above the text box, and then click on 'Add new field'. A new field will appear in the grid.

1.5. Remove a field in your database

Select a custom database.
Click on the 'Remove this field' button, select the field you want to remove, and then click 'OK'.

1.6. Fill a database field

Select a custom database.
Type a value on the text box near the button 'Fill field'.
Click on the 'Fill field' button, select the field you want to fill then click 'OK'.

1.7. Fill a custom database

Right click on the grid then select 'add line' or 'add multiple lines'.

1.8. Save and Undo changes

Quotes and custom databases data can be modified manually in the grid.
You have to click on 'Save' to save the new data.

Click on 'Undo changes' button to undo all your previous changes.

1.9. Export data

The 'Export to CSV' button is used to export the current grid data to CSV.

[TOP]

1.2.4.2 Application Objects

Application Objects

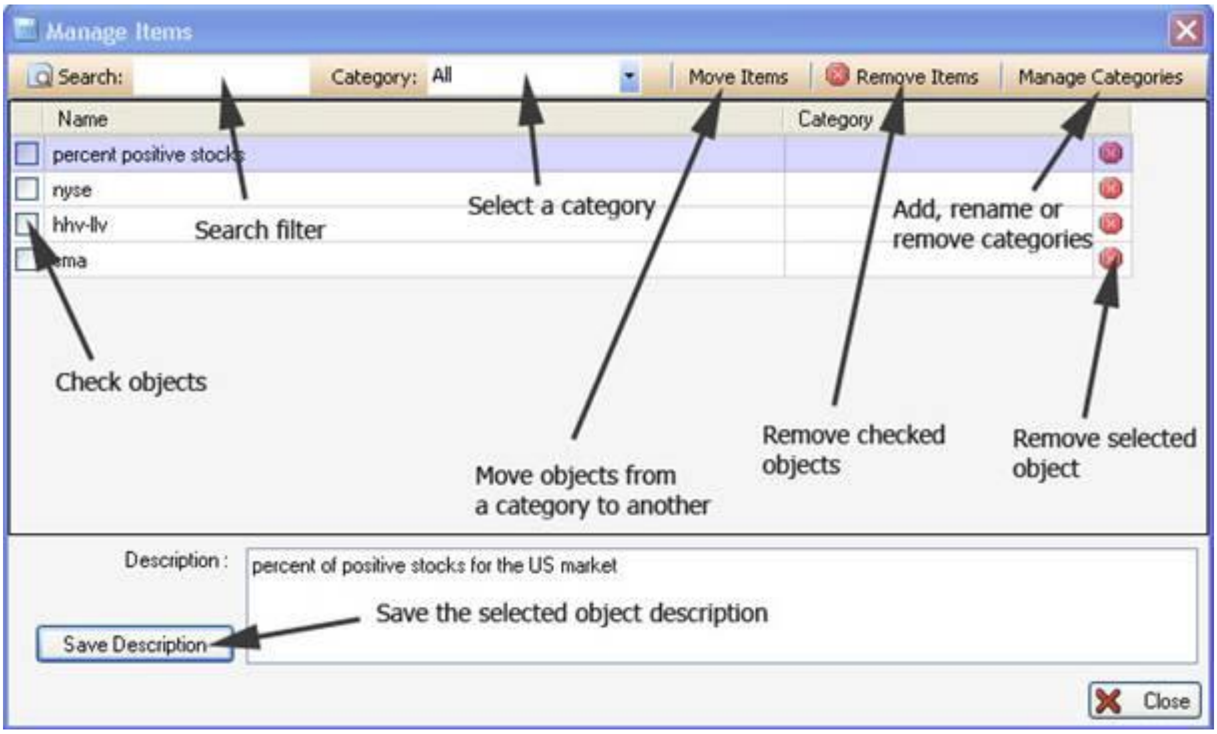
Introduction

We use the word 'object' or 'item' to point out:

- Trading systems
- Watch-list
- Composite indexes
- List of rules
- List of symbols
- Simulator metrics
- ...

All these objects share some common controls.

These controls are used to manage, save and load these objects.



Category: All

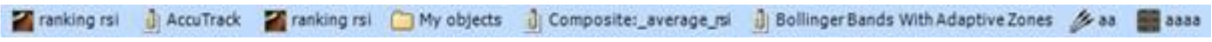
	Name	Category
□	percent positive stocks	
□	nyse	
□	hhv-llv	
□	ema	

Description : percent of positive stocks for the US market

Each object is stored in a specific file.
To rename an object, double click on its name, and then type a new name.
An object can either belong to a category or not.
These objects can be shared, you can visit www.quantshare.com for a list of all shared objects.

The majority of the plug-ins have drag & drop features that let you drag objects and drop them on widget or bookmark panels.

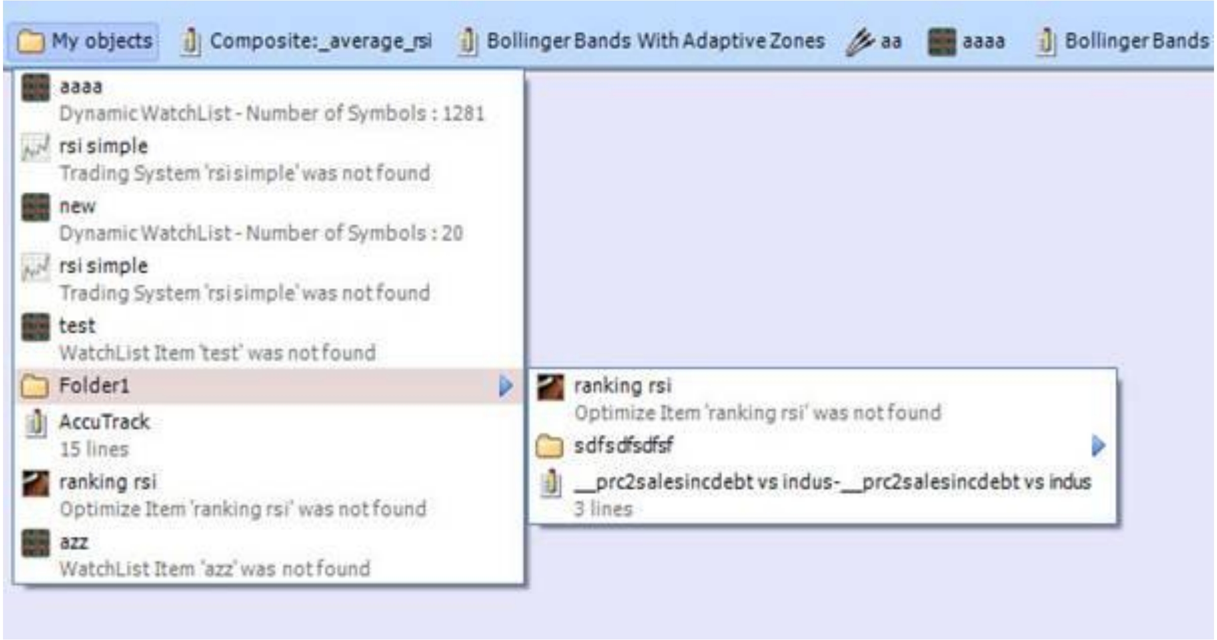
Bookmark panel



This panel hosts folders and objects (generally shortcuts of objects are used)

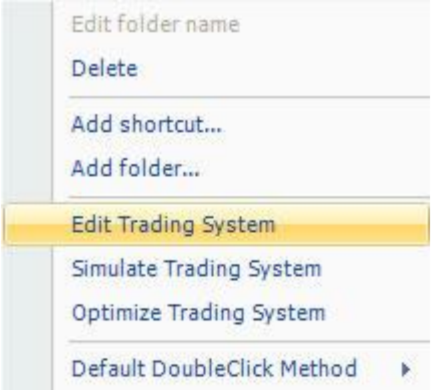
Right click on the bookmark panel to open a menu. Using this menu, you can create a new folder, add an object or object shortcut, update a folder name...

You can drag and drop objects within the bookmark panel to change their locations or to move them from a folder to another.


My objects
Composite:_average_rsi
Bollinger Bands With Adaptive Zones
aa
aaaa
Bollinger Bands
aaaa
Dynamic WatchList - Number of Symbols : 1281
rsi simple
Trading System 'rsi simple' was not found
new
Dynamic WatchList - Number of Symbols : 20
rsi simple
Trading System 'rsi simple' was not found
test
WatchList Item 'test' was not found
Folder1
AccuTrack
15 lines
ranking rsi
Optimize Item 'ranking rsi' was not found
azz
WatchList Item 'azz' was not found
ranking rsi
Optimize Item 'ranking rsi' was not found
sdfsdfsdfsf
__prc2salesincdebt vs indus-__prc2salesincdebt vs indus
3 lines

Right click on any object to display a menu. The menu content is specific to each object type.

Example: 'Trading system' object menu.


Edit folder name
Delete
Add shortcut...
Add folder...
Edit Trading System
Simulate Trading System
Optimize Trading System
Default DoubleClick Method

[TOP]

1.2.5 QuantShare Language

1.2.5.1 QuantShare Language

1. QuantShare Language

1.1. Introduction

QuantShare language is a simple, complete and fast scripting language.
It is a vector based language and contains more than 200 build-in functions and indicators.

Example of a vector-based language: if you type 'a = 1'.
The variable 'a' will be represented as an array of N elements (N represent the number of bars for a specific symbol) where each element of the array has a value of '1'.

Other examples:
c = close;

This is the representation of the 'c' variable.

10.5	11	11.2	11	10.8	10.5	10	10	10.3	11.2

The current symbol has 10 quotes (elements).
The first column represents the first bar or bar number '0'.

If we type now: c = close + 2;

The representation of the 'c' variable will be:

12.5	13	13.2	13	12.8	12.5	12	12	12.3	13.2

We added "2" to each element of the array.
You can see these arrays at any time using the debugger tool. (Right click on the editor).

In the next example, we will use a very common function 'ref', this function let you refer to old or future quotes.

c = ref (close, 1) makes the array shift by one bar to the right.

NA	10.5	11	11.2	11	10.8	10.5	10	10	10.3

'NA' means that the element of the array contains no data.

1.2. Basic Concepts

A variable can be a number or a string.

v1 = "test"; (assign the text 'test' to the variable 'v1').
v2 = 2; (assigns the number 2 to the variable v2).

You cannot do math calculation using both numeric and text variables.

v3 = v2 + 2; (Correct)
v3 = v2 + v1; (Not Correct)

To concatenate two string variables or a string and a numeric variables: use "." (Point)

v3 = "test ".v2." test";

There are two types of functions:

- Void function or function that does not return a value
  Example: 'Plot' function plots a time-series on a chart.

- Functions that returns a numeric or a string array
  Example: 'Rsi' (Relative strength index indicator) returns a numeric array.

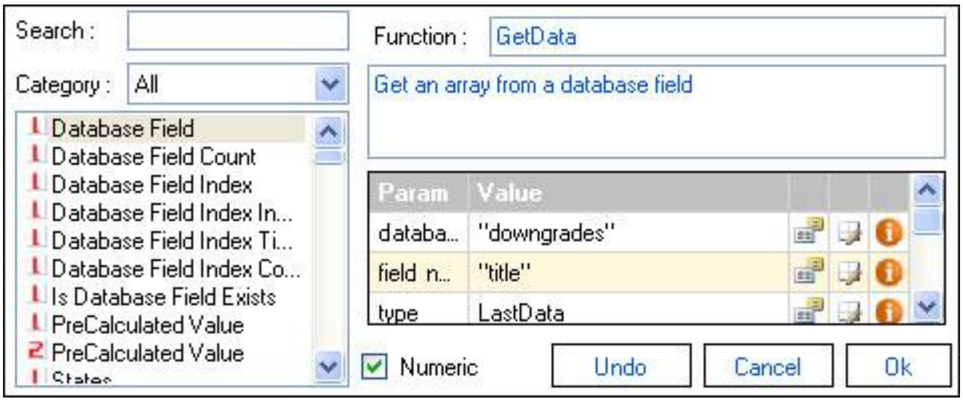
1.3. Edit a formula

Click on 'New Chart' in the 'View' menu to open a new chart.
Right click on the chart and click on 'Edit Formula'; the formula editor will open.

1.4. Add functions

There are two ways to add a function:

- You can add a function by typing it in the formula editor.
- You can add a function by clicking on the 'Add void' or 'Add function' buttons; the 'Function selector' form will appear.



Search :

Category : All

1 Database Field
1 Database Field Count
1 Database Field Index
1 Database Field Index In...
1 Database Field Index Ti...
1 Database Field Index Co...
1 Is Database Field Exists
1 PreCalculated Value
2 PreCalculated Value
1 States

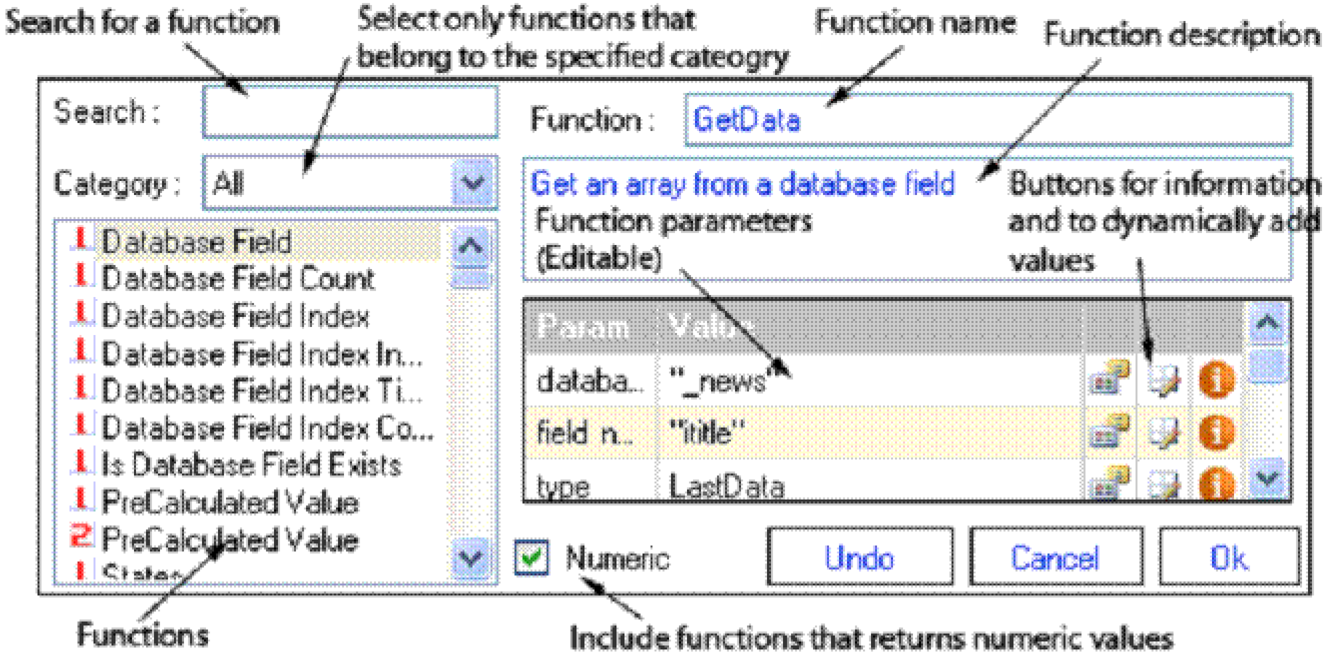
Function : GetData

Get an array from a database field

Param	Value
databa...	"downgrades"
field n...	"title"
type	LastData

☑ Numeric

1.5. Function selector



Search for a function

Select only functions that belong to the specified cateogry

Function name

Function description

Search :

Category : All

Function : GetData

Get an array from a database field

Function parameters (Editable)

Buttons for information and to dynamically add values

Param	Value
databa...	"_news"
field n...	"title"
type	LastData

1 Database Field
1 Database Field Count
1 Database Field Index
1 Database Field Index In...
1 Database Field Index Ti...
1 Database Field Index Co...
1 Is Database Field Exists
1 PreCalculated Value
2 PreCalculated Value
1 States

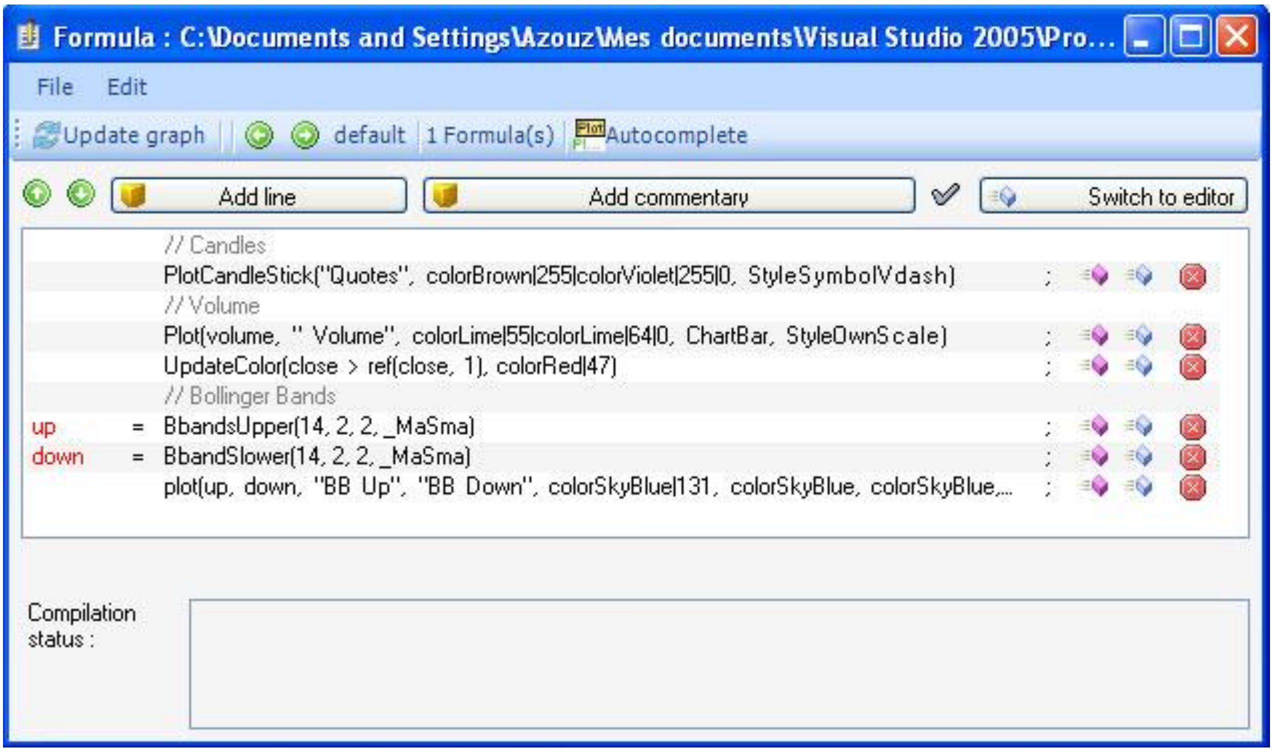
☑ Numeric

Functions

Include functions that returns numeric values

1.6. Using the wizard

Click on 'Switch to wizard', in the formula editor form.



Formula : C:\Documents and Settings\Azouz\Mes documents\Visual Studio 2005\Pro...

```
          // Candles
          PlotCandleStick("Quotes", colorBrown|255|colorViolet|255|0, StyleSymbolVdash)   ;
          // Volume
          Plot(volume, " Volume", colorLime|55|colorLime|64|0, ChartBar, StyleOwnScale)   ;
          UpdateColor(close > ref(close, 1), colorRed|47)                                  ;
          // Bollinger Bands
up    =   BbandsUpper(14, 2, 2, _MaSma)                                                     ;
down  =   BbandSlower(14, 2, 2, _MaSma)                                                     ;
          plot(up, down, "BB Up", "BB Down", colorSkyBlue|131, colorSkyBlue, colorSkyBlue,...  ;
```

Compilation status :

There are five columns in the wizard:

1. Variable name (Red)
2. Line of code
3. Button to add 'Void' functions
4. Button to add 'Return' functions
5. Button to remove a line

You can type '//' in the second column to specify to the compiler that this line is a commentary line.
Switch to the normal editor by clicking on 'Switch to editor' button.

1.7. Functions to plot time-series

Here is the list of the functions that let you plot data:

- Plot: plots a vector; you can specify the color, the graph type...
- Plot1: plots two vectors, example: higher and lower Bollinger bands.
- PlotCandleStick: plots an OHLC chart
- PlotFixed: plots a fixed chart, example: average performance per day of the week or volatility for the next ten days per Relative Strength Index values
- PlotSymbol: plots symbols above or below other graphs.
- SetMaxScale: sets the Y-Axis maximum value.
- SetMinScale: sets the Y-Axis minimum value.
- SetScale: sets the Y-Axis maximum and minimum value.
- UpdateColor: updates a graph color, on certain X-Axis values.

1.8. Update formulas dynamically

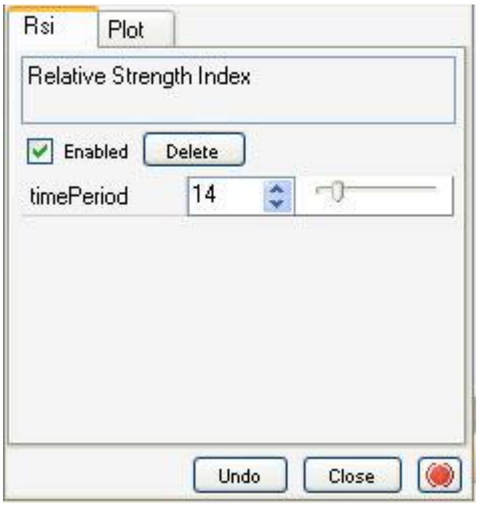
Move the mouse over a graph then right click, a small form will appear.
Tabs shown on that form refers to functions available in your formula.

If for example your formula is:

```
r = rsi (14);
Plot(r, "RSI", colorRed, ChartLine, StyleOwnScale);
```

Two tabs will be available (Rsi and Plot).
Under Rsi tab there is a cell whose name is 'timePeriod'; this cell has a value of "14".

Click on that value, and update it using the numeric text box or the track bar.
See how the graph updates dynamically while you change this value.

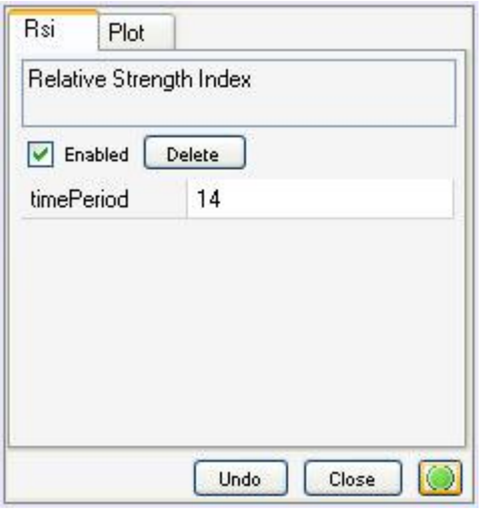


Relative Strength Index

☑ Enabled

timePeriod 14

Click on the green button to make the form stick.



☑ Enabled

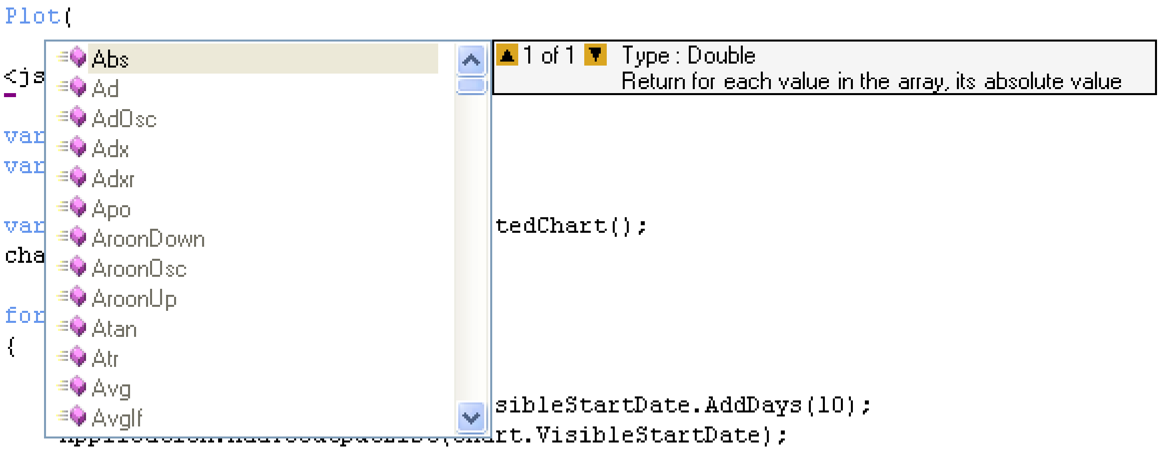
Relative Strength Index

timePeriod 14

1.9. AutoComplete tool

In the 'Formula Editor', press on CTRL+SPACE, a list of 'Void' functions appears.

Now type something like 'a = ', then press on CTRL+SPACE, another list appears; this list contains all 'Return' functions.



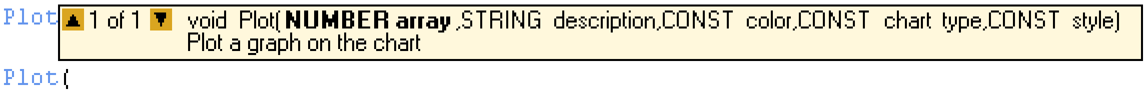
```
Plot(

<js
var
var

var                                                 tedChart();
cha

for
{
                                                    sibleStartDate.AddDays(10);
                                                    rt.VisibleStartDate);
```

▲ 1 of 1 ▼ Type : Double
Return for each value in the array, its absolute value

There is also an 'AutoComplete' button on the 'Formula Editor'; click on that button, then type in something in the editor, the 'AutoComplete' list now appears each time you type in a letter.

1.10. Tooltip information

Tooltip shows you information about functions.
There are two kinds of Tooltip; one that appear right after you type in '(', example: 'Plot ('.
The other Tooltip appear when you move the mouse over a function.

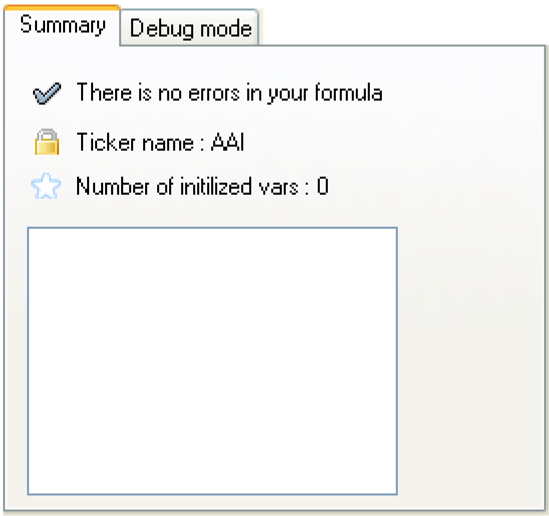


```
Plot(
Plot(
```

▲ 1 of 1 ▼ void Plot(**NUMBER array**,STRING description,CONST color,CONST chart type,CONST style)
Plot a graph on the chart

1.11. Debugger

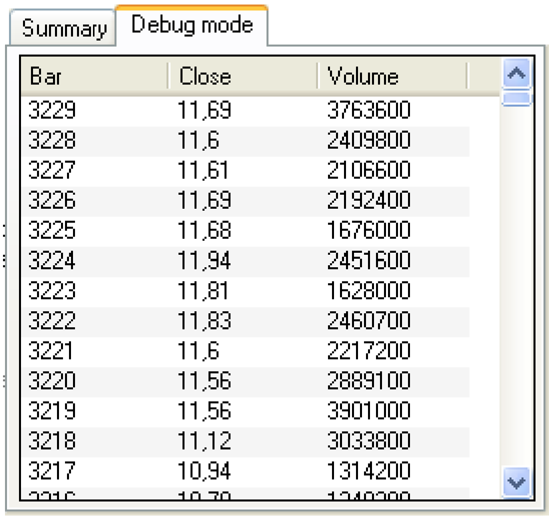
The debugger displays the elements of the variables initialized within the formula.
In the 'Formula Editor', right click on the editor.



✓ There is no errors in your formula

Ticker name : AAI

Number of initilized vars : 0

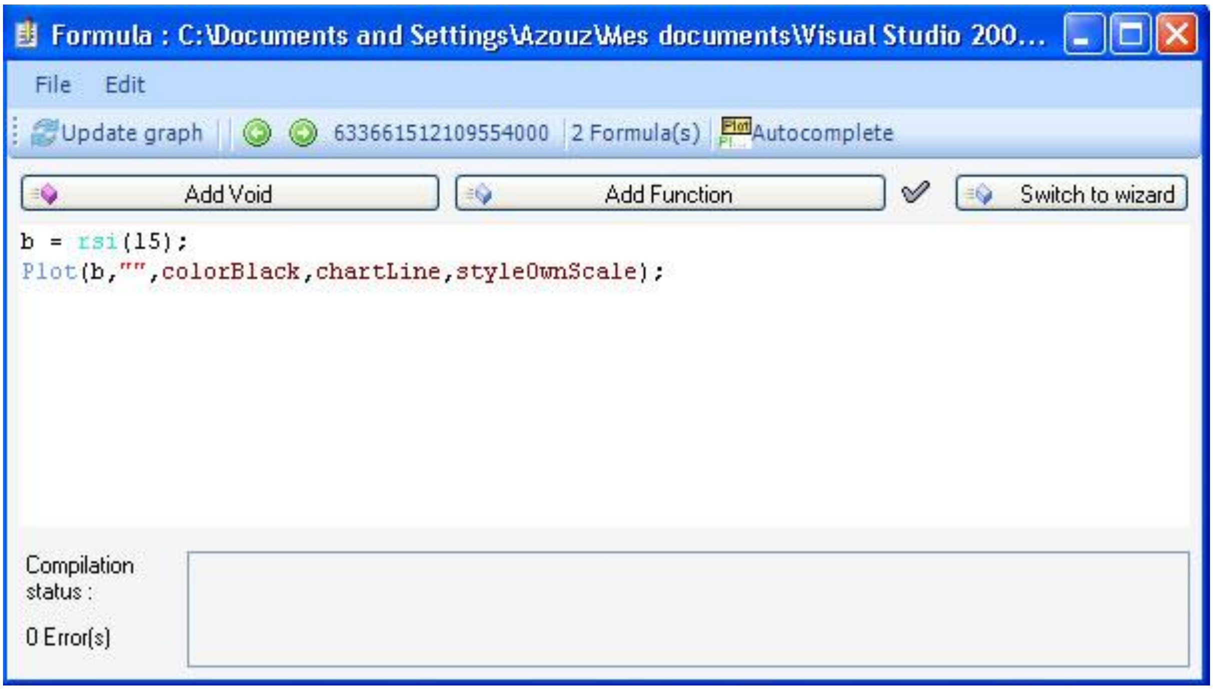


Bar	Close	Volume
3229	11,69	3763600
3228	11,6	2409800
3227	11,61	2106600
3226	11,69	2192400
3225	11,68	1676000
3224	11,94	2451600
3223	11,81	1628000
3222	11,83	2460700
3221	11,6	2217200
3220	11,56	2889100
3219	11,56	3901000
3218	11,12	3033800
3217	10,94	1314200

1.12. Formula errors

When typing a formula, an icon located at the top shows you whether your formula syntax is correct or not, clicking on that icon shows you the formula errors if any.
The icon is red when there is an error otherwise it is blue.

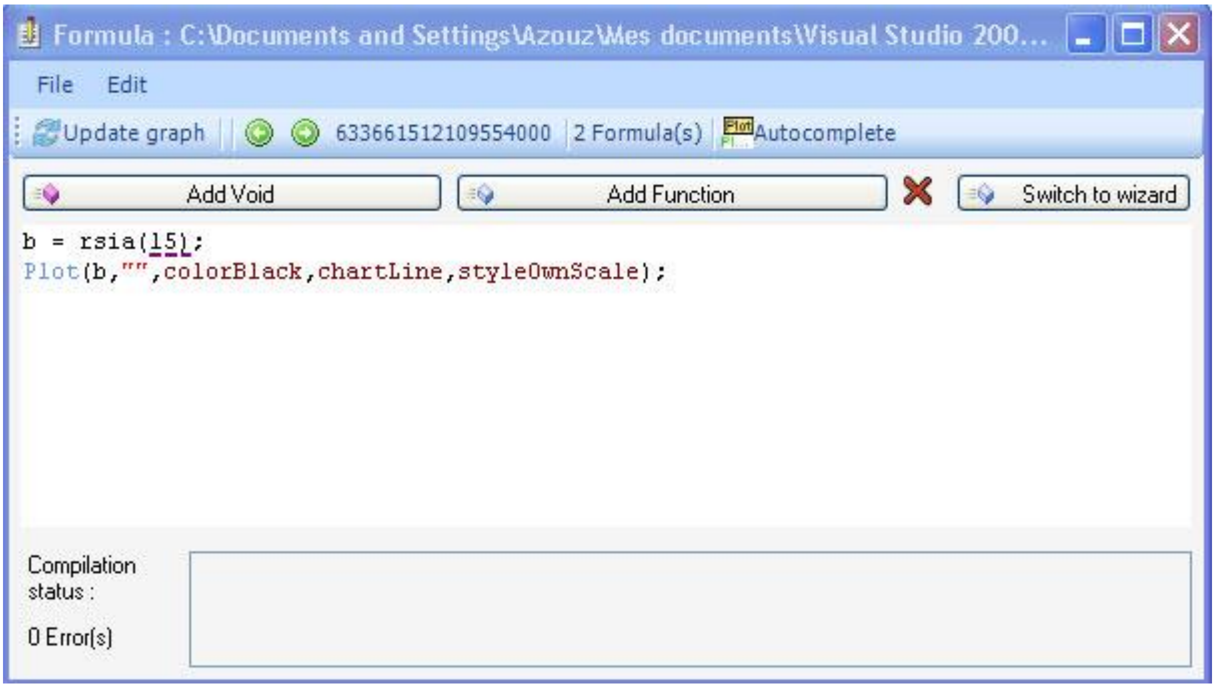
The location of the error is underlined with a dotted line.



```
b = rsi(15);
Plot(b,"",colorBlack,chartLine,styleOwnScale);
```

Compilation status :

0 Error(s)



```
b = rsia(15);
Plot(b,"",colorBlack,chartLine,styleOwnScale);
```

Compilation status :

0 Error(s)

After compiling the formula by clicking on 'Update Graph', the errors if any will be listed on the 'Compilation status' text box.

[TOP]

1.2.5.2 Advanced Rules

1. Advanced Rules

1.1. Searchfor

Used to detect any complex pattern
Used to create advanced trading rules

Let you create unrestrictive rules

Syntax:

Searchfor syntax1 **then** syntax1 **for (**rule**) set (**special_rule**)**

syntax1: special_rule **after** *period* **within** *period* **all**

special_rule: Special rule (Supports QuantShare functions + special functions)
rule: Normal rule (Any rule that could be created using the QuantShare language)

Explanation: this function looks for successive events.
Example: search for a stock that rises 20 percent within 30 bars, then drops by 10 percent for the next 10 – 20 bars.

After (optional)**:** specify that **special_rule** will be checked only after the number of bars specified after the keyword **after**

Within (optional)**:** specify that **special_rule** must be TRUE in at least one of the bars starting from 'current bar' + 'after value' to 'current bar' + 'after value' + 'within value'

All (optional)**:** specify that **special_rule** must be TRUE for all the bars starting from 'current bar' + 'after value' to 'current bar' + 'after value' + 'within value'

For (optional)**:** calculates the pattern only when the rule provided in (**for**) keyword is TRUE

Set (optional): The **Searchfor** function returns 1 on bars where the pattern is found, if we specify a formula in (**set**) keyword, the **Searchfor** function will returns the result of this function instead of 1

Special functions (that could be used in **special_rule**):

- _perf (*Array*): Gets the percent of change from the specified *array,* from the bar where the previous condition occurred to the current bar.
- _perf(*Array, lag, length*): Gets the percent of change from the specified *array*, from "the current bar minus lag minus length" to "the current bar minus lag".

- _change (*Array*): Gets the raw change from the specified *array,* from the bar where the previous condition occurred to the current bar.
- _change (*Array, lag, length*): Gets the raw change from the specified *array*, from "the current bar minus lag minus length" to "the current bar minus lag".

- _avg (*Array*): Gets the average value of the specified *array*, from the bar where the previous condition occurred to the current bar.
- _avg(*Array, lag, length*): Gets the average value of the specified *array*, from "the current bar minus lag minus length" to "the current bar minus lag".

- _min (*Array*): Gets the minimum value of the specified *array*, from the bar where the previous condition occurred to the current bar.
- _min (*Array, lag, length*): Gets the minimum value of the specified *array*, from "the current bar minus lag minus length" to "the current bar minus lag".

- _max (*Array*): Gets the maximum value of the specified *array*, from the bar where the previous condition occurred to the current bar.
- _max(*Array, lag, length*): Gets the maximum value of the specified *array*, from "the current bar minus lag minus length" to "the current bar minus lag".

- _sd (*Array*): Gets the standard deviation value of the specified *array*, from the bar where the previous condition occurred to the current bar.
- _sd (*Array, lag, length*): Gets the standard deviation value of the specified *array*, from "the current bar minus lag minus length" to "the current bar minus lag".

- _ref (*Array , index_lag*): Gets a value from the specified *array* on the bar where the condition "*last condition index minus the specified index_lag"* occurred.

  Index equals to 0 means get a value from the specified array on the bar where the last condition occurred.

  Index equals to 1 means get a value from the specified array on the bar where the next to last condition occurred.

  FRef (*Array*): same as the last function but uses zero as "*index_lag*"

- _dis(*index_lag*): Gets the distance (in bars) between the current bar and the bar where condition "*last condition index minus the specified index_lag"* occurred.

Index equals to 0 means get the number of bars between the current bar and the bar where the last condition occurred.

- _iff (*condition, if_true, if_false*): Evaluates and returns the result of the **if_true** formula if the condition is TRUE, otherwise evaluates and returns the result of the**if_false** formula.

- _slope (*Array*): Gets the slope of the line that starts at the array value for the bar where the previous condition occurred to the array value of the current bar.
- _slope (*Array, lag, length*): Gets the slope of the line that starts at the array value for "the current bar minus lag minus length" to the array value of "the current bar minus lag".

- _aboveline (*Array*1, Array2): Returns TRUE if all the elements of the Array2 between the bar where the previous condition occurred to the current bar are ABOVE the line that starts at the Array1 value for the bar where the previous condition occurred to the Array1 value of the current bar.
- _aboveline (*Array1, Array2, lag, length*): Returns TRUE if all the elements of the Array2 between"the current bar minus lag minus length" and "the current bar minus lag" are ABOVE the line that starts at the Array1 value for "the current bar minus lag minus length" to the Array1 value of "the current bar minus lag".

- _belowline (*Array*1, Array2): Returns TRUE if all the elements of the Array2 between the bar where the previous condition occurred to the current bar are BELOW the line that starts at the Array1 value for the bar where the previous condition occurred to the Array1 value of the current bar.
- _belowline (*Array1, Array2, lag, length*): Returns TRUE if all the elements of the Array2 between"the current bar minus lag minus length" and "the current bar minus lag" are BELOW the line that starts at the Array1 value for "the current bar minus lag minus length" to the Array1 value of "the current bar minus lag".

Notes:
Special functions start with "_"

_change(close) is equivalent to _change(close, 0, _dis(0))

Example:

Let us say we want to look for stocks that have the following pattern:

- Positive performance of at least 20 percent in the last 30 bars

- Then a negative performance of at least -10 percent in a period between 10 and 50 bars, during this period the stock close price must be lower than the price at which the last condition occurred
- Then a positive performance of at least 20 percent within the next 30 bars

Here is the formula that describes this pattern:

```
rule = Searchfor perf(close, 30) > 20
then
_perf(close) < -10 && _max(close) <= _ref(close)
after 10 within 50
then
_perf(close) > 20
within 30;
```

Explanation:

- Positive performance of at least 20 percent in the last 30 bars. "Perf" is a vector-based function, while "_perf" is a special function.

  ```
  perf(close, 30) > 20
  ```

- Then a negative performance of at least -10 percent in a period between 10 and 50 bars, the **_perf** functions doesn't need a period parameter as **perf** do, because it will use the (current bar - last condition bar) as the period parameter

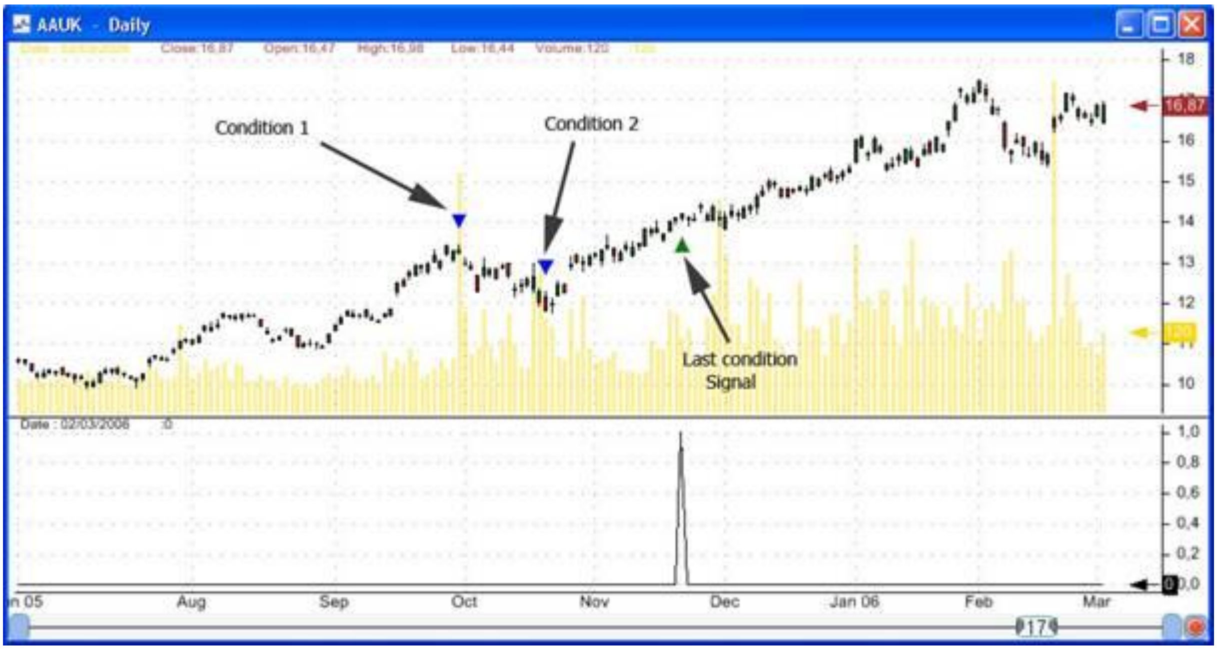
  ```
  _perf(close) < -10
  ```

  during this period, the stock close price must be lower than the price at which the last condition occurred. Note the difference between **close** and **_ref(close)**. **Close** references the close price for the current bar, while **_ref(close)** references the close price of the last condition bar. **_ref(close, 1)** references the close price of the next to last condition bar if it exists (otherwise it takes the last condition bar). Now, we only have one condition (**perf(close, 30) > 20**), because the current condition is not evaluated yet.

  ```
  _max(close) <= _ref(close)
  ```

- Then a positive performance of at least 20 percent within the next 30 bars.

  ```
  _perf(close) > 20 within 30
  ```



The first and the second panes contain the same 'Searchfor' formula.
They do not look the same because if a 'Searchfor' rule is plotted on a pane where a candlestick chart is available then the signal and conditions will be shown as triangles marks.

1.2. Output

Used to simulate outputs

References future bars

Could be used to create custom outputs when analyzing trading rules or when creating prediction models

Syntax:

Set syntax1 **where** syntax1 **then** syntax1 **for (**rule**)**

syntax1: special_rule **after** *period* **within** *period* **all**

special_rule: Special rule (Supports QuantShare functions + special functions) (look at **Searchfor** paragraph)
rule: Normal rule (Any rule that could be created using the QuantShare language)
(**then** syntax1) is optional.

Special rules are the same as in **Searchfor**.

Example:
Select ‘Analysis’ in menu, then click on ‘Rules Manager’.
Select or create some rules then click on ‘Analyze’.
Select ‘Outputs’ on the left panel, and then click on ‘Select Outputs’ on the right panel.
On the new ‘Output’ form, select ‘Custom output’ in the list, and then click on ‘Add’.
On the right panel, select the newly created output and then click on ‘Edit’.
Click on ‘click here to select an example’ and choose ‘… hedging with an option put’

This formula simulate an entry with a simultaneous buy of an option put.
In the **Where** input, **_dis(0) >= 60**, means that we evaluate the output with the condition that the number of bars equals to 60, in other words we exit after 60 bars.
In the **Set** input, we used a conditional statement to simulator the hedging strategy.
If the instrument return is higher than -20 percent then we want to set the return minus the cost of the option (2 percent) as output.
If the instrument return is lower than -20 percent then the output will be -20 percent (the maximum we can lose).
The _perf(close) calculates the performance of the instrument from the current bar to the bar where the last condition occurred, and in this example the last condition is **_dis(0) >= 60**, so we get the return of the next 60 bars.

This example is just a simulation of a hedge strategy and assumes:

The option expires in 60 bars
The option costs is 2 percent of the instrument price
The option strike is at -20 percent of the current instrument price

(If the price is 100$, the strike is at 80$, and the option price is 2$)

1.3. Controls

Right click on a pane, and then select ‘Edit formula’

Type CONTROL+SPACE, then select ‘SEARCHFOR...THEN’ or ‘SET...WHERE...’ to have a form appears.

You can type your rules directly in the inputs provided, or you can check the example provided.
Click on ‘Disabled’ and select an item to enable inputs.
When done, click on ‘OK’, your formula should appear in the formula editor.

1.4. Additional keywords

NoJump: This keyword tells the application not to jump after a pattern is discovered.
The default behavior (for speed purposes) is to jump to the bar where a pattern is discoverd, and to continue the search beginning from that bar.
Example: Starting from bar 5, the "searchfor" function discovered a pattern at bar 50.
The next search will begin at bar 51. However, if "nojump" were specified then the search would start at bar 6.
What you need to know is that using "nojump" keyword enables the application to discover more patterns. However, the drawback is that the function will be much slower.
The "NoJump" keyword works only with "Searchfor". The "Set" advanced rule uses the "nojump" automatically. This is because this function must return a value for each bar.

Example of usage: (`nojump` must be placed just after the `SEARCHFOR keyword`)

```
var1 = SEARCHFOR nojump gapup() THEN _perf(close) > 20 after 10 within
100;
```

Depth: This function tells the application to perform deep analysis of a particular condition. This will sometimes lead to the discovery of more patterns. By default, the application applies a depth of one.

Example:

```
var1 = SEARCHFOR gapup() THEN close > ref(close, 1) after 10 within 100
depth(5) THEN gapup() depth(10);
```

[TOP]

1.2.6 Plug-ins

1. Plug-ins

A Plug-in is a program that interacts with the main program and brings him more functionality.

This is the list of the currently available plug-ins:

- Indicators
- ASCII Importer
- Downloader
- News
- Divers
- Task Manager
- Script Manager
- Watch list
- Custom functions
- Composites
- Widget Panel
- Screener
- Rules Manager
- Ranking System Manager
- Simulator
- Advanced Money Management
- Artificial Intelligence
- Optimizer
- Portfolio123
- Sharing Server

[TOP]

1.2.6.1 Indicators

1. Indicators

1.1. Introduction

More than 100 indicators are available:

- Relative Strength Index
- Aroon Oscillator
- Bollinger Bands
- Average True Range
- Absolute Price Oscillator
- On Balance Volume
- ...

1.2. Open the indicator form

Click on 'Indicators' in the 'View' menu.
Select a category, and then select an indicator.
Right click on that indicator and a menu will show up.
On that menu, you can edit the indicator files, rename them, create new ones and set the action to take when you double click on them.

1.3. Drag & Drop indicators

In the indicator form, click on an indicator then drag & drop it to a chart.
You can also drag any pane formula and drop it into another pane.

1.4. Add an indicator to a chart

Besides adding indicators by updating the chart formula, you can add indicators using the indicator form by either double clicking on an indicator or drag & drop the indicator to a chart.
You can also right click on a chart, select 'Indicators' then click on 'Add study'.

1.5. Create your own indicator

You can create your own indicator using the 'Custom functions' plug-in.

[TOP]

1.2.6.2 Custom functions

1. Custom functions

1.1. Introduction

This plug-in lets you create custom function that you can use in your formulas.
Functions are created using CSharp or JScript.Net languages.

Let us say you created a function which have two numeric parameters and whose name is 'MyFunction'.
To use this function, open the formula editor (Right click on a chart, then click on 'Edit formula'), type for example 'a = ', then click on CTRL+SPACE; in the list box you will find your function 'MyFunction'.

1.2. Create a new function

Open the 'Create functions' form by clicking on 'Create functions' in the 'Tools' menu.

In the 'Create functions' form, click on 'Add', type a name then save.

Select the function, you have just created, in the 'Functions' panel, then move to the 'Script' panel.

Press on 'CTRL+SPACE' to see the initialized variables.

The application creates a variable for each parameter.
'result' is the variable that your function will return to the application
'result' is of type 'VectorD' if your function returns a numeric time-series or 'VectorS' if your function return a string time-series

Example: to create a function that returns an array containing the value '2'.

Type this code in your script panel:

```
for(var i: int = 0;i<result.Length;i++)
{
  result.SetValue(i, 2);
}
```

You can add parameters to your function, by clicking on 'Add a parameter'.
You can update the parameters name, type, default value and description.
After adding a parameter, press on CTRL+SPACE to see the new variable in the list.

Example: to create a function that adds one to the provided parameter and returns the new array.

```
var val : double;
for(var i: int = 0;i<result.Length;i++)
{
  val = var1.GetValue(i);
  result.SetValue(i, val + 1);
}
```

When done, click on Save.

1.3. Update a function

Clicking on 'Save' button will compile the function and updates it.

1.4. Remove a function

Select a function, click on 'Remove' then confirm.

1.5. Functions

The 'cFunctions' class contains the symbol name, close, open, high, low, volume, open interest and date vectors of the symbol that is used when executing a formula.
Example: To get the close vector, type:

```
var close : VectorD = cFunctions.Close;
```

Create an empty vector
To create an empty numeric vector, type:

```
var vectorNumeric : VectorD = cFunctions.CreateNumericVector();
```

To create an empty text vector, type:

```
var vectorText : VectorS = cFunctions.CreateTextVector();
```

This class also contains three important functions:

CompileFormula: This function compiles a QuantShare formula and returns a QSFormula class. This class can be used to extract vectors.
Example: `cFunctions.CompileFormula("a = rsi(14);").GetVectorDouble("a");`

GetCustomDatabaseData: This function loads custom database data for the current symbol or another one and returns a VectorCustom class.
The vectorCustom class is like the other vector classes with the difference that it can contain more than one element in a bar period.
If you execute a function on a daily timeframe and you reference an intraday database (news for example), then for a specific bar (date), you may have more than one news item.

SetForwardAndBackwardBars: This function let you define how many backward and forward bars your formula use.
Forward and Backward bars are used to optimize the execution of formulas.
As an example, the RSI or relative strength index function (rsi(14)), uses 14 backward bars and 0 forward bars.
Because for each bar, it needs the previous 14 bars to perform its calculation and it does not require any future bar.

[TOP]

1.2.6.3 Composite

1. Composite

1.1. Introduction

Create any kind of composite using this plug-in.

Examples of composites:

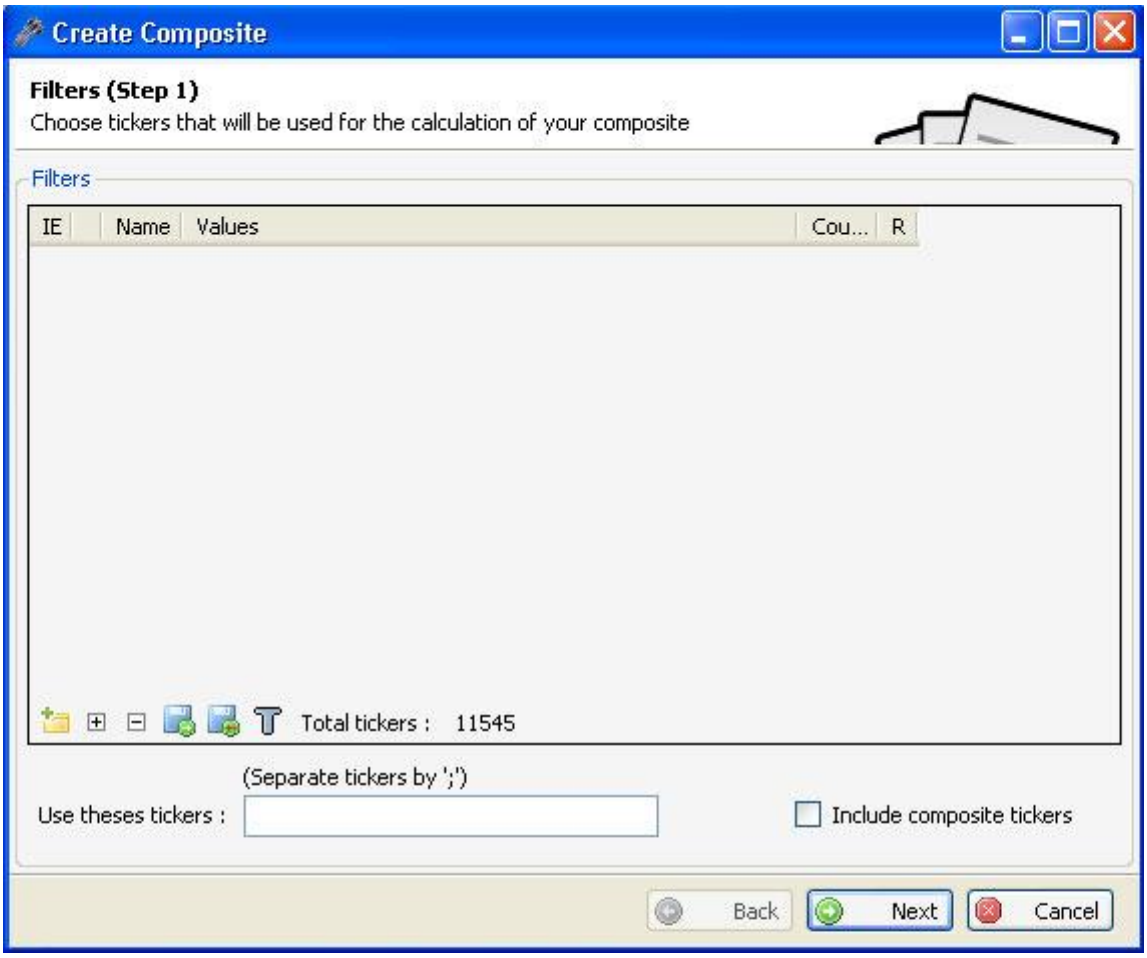
- Percentage of stocks trading above their 50-day moving average
- Number of advancing stocks within a sector
- Price to book ratio of the entire US stock market
- Percentage of stocks that have an overbought RSI

The composite plug-in automatically detect if there are new quotes and recalculate the composites if necessary.

1.2. Create a composite

Tools -> Composite

Click on 'Add' to open a new form.



Create Composite

Filters (Step 1)

Choose tickers that will be used for the calculation of your composite

Total tickers : 11545

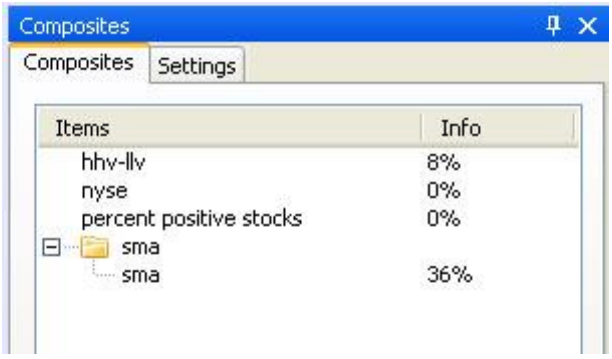
Use theses tickers : (Separate tickers by ';')

☐ Include composite tickers

Select the symbols you want to include in the composite calculation.
'Include composite tickers' check box is used to specify whether to include composite symbols or not.

N.B: If there are two composites which have this box checked, a calculation error will appears in the case where composite 'A' symbols include composite 'B' and composite 'B' symbols include composite A.

Type then the composite formula, choose the calculation function, the start and end periods, then the composite name, full name and description.
The new composite will be added to the 'Composite' form's grid and the calculation will start.



Items	Info
hhv-llv	8%
nyse	0%
percent positive stocks	0%
⊟ sma	
sma	36%

1.3. Update a composite

In the 'Composite form', select a composite item and then click on 'Update' button.

1.4. Remove a composite

In the 'Composite form', select a composite item and then click on 'Remove' button.

1.5. Re-Calculate a composite

In the 'Composite form', select a composite item and then click on 'Recalculate' button.
To recalculate all the composites, click on 'Recalculate all' button.

1.6. Composite Plug-in Settings

In the 'Composite' form, select the 'Settings' tab.
You can select whether the composite should be recalculated automatically on new data or not.

[TOP]

1.2.6.4 Watch list

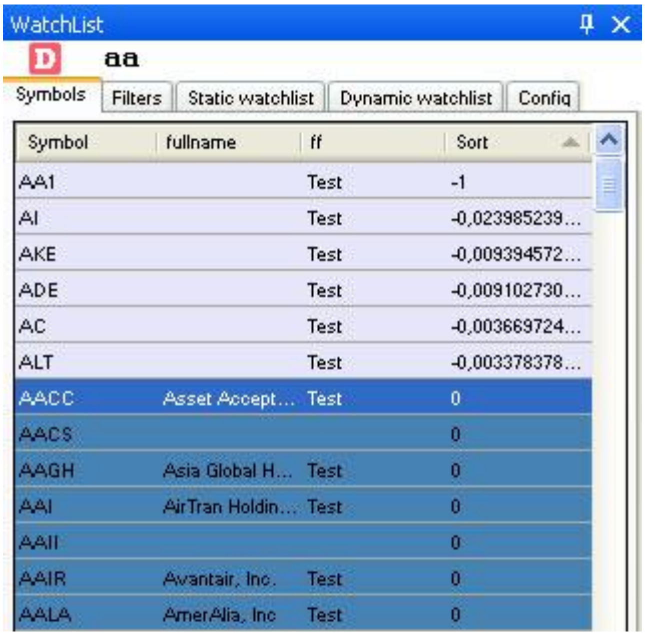
1. Watch list

1.1. Introduction

This plug-in displays symbols that meet your criteria.
Watch lists are updated dynamically; list of symbols contained in the watch lists automatically changes on new quotes or new databases data.

To open the watch list plug-in:

- Click on menu 'Tools'
- Click on item 'Watch List'



Symbol	fullname	ff	Sort
AA1		Test	-1
AI		Test	-0,023985239...
AKE		Test	-0,009394572...
ADE		Test	-0,009102730...
AC		Test	-0,003669724...
ALT		Test	-0,003378378...
AACC	Asset Accept...	Test	0
AACS			0
AAGH	Asia Global H...	Test	0
AAI	AirTran Holdin...	Test	0
AAII			0
AAIR	Avantair, Inc.	Test	0
AALA	AmerAlia, Inc	Test	0

1.2. Create a simple watch list

A simple watch list is a watch list where you can specify simple criteria, example: symbol names, symbol market...

To create a simple watch list:

- Click on 'Filters' tab in the 'Watch-list' form
- Create your filter
- Click on 'Show'

1.3. Create a static watch list

A static watch list is a watch list that contains a list of symbols.

To create a static watch list:

Click on 'watch-lists' tab in the 'Watch-list' form

- Type the symbols

Click on 'OK'

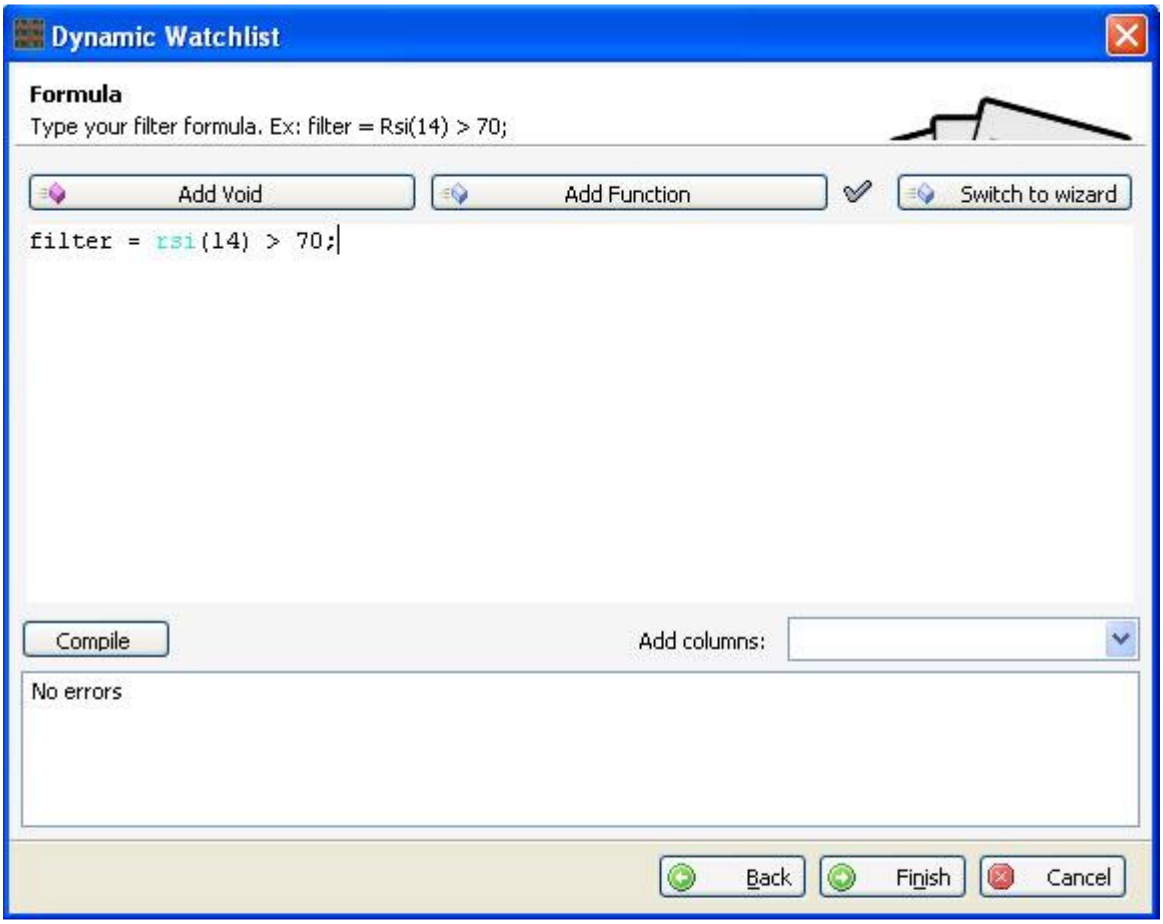
If you want to create a static watch list from the active watch list, just click on 'Create from active watch list' button after typing a watch list name.

1.4. Create a dynamic watch list

A dynamic watch list is a watch list where you can specify complex criteria.
Dynamic watch-lists update its list of symbols dynamically on new data.

To create a dynamic watch list:

- Click on 'watch-lists' tab in the 'Watch-list' form
- Click on 'create dynamic watchlist'
- In the new form, set the filter
- Click on 'Next'
- Type your formula and make sure it is correct by clicking on 'Compile'
- Click on 'Finish'

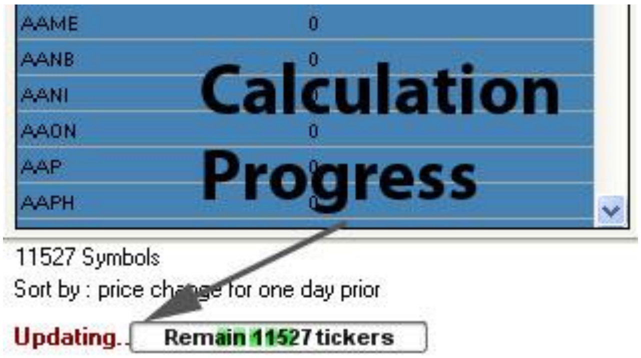


Type your filter formula. Ex: filter = Rsi(14) > 70;

```
filter = rsi(14) > 70;
```

No errors

After clicking on "Finish", the watch list plug-in will display all the symbols and start the calculation process.



AAME	0
AANB	0
AANI	
AAON	0
AAP	0
AAPH	0

11527 Symbols
Sort by : price change for one day prior
Updating.. Remain 11527 tickers

When the calculation process is done, symbols that fail to pass the criteria are removed.
The dynamic watch list will calculate the variable 'filter' for each symbol and if this variable is true or different from zero then it will include the symbol.

Example: Select overbought symbols (Relative strength index superior to 70)
filter = rsi(14) > 70;

1.5. Add columns to a dynamic watch list

You can add column to a dynamic watch list, the watch list grid will shows you these columns with their values for each symbol.
To add a column, use the function 'AddColumn' or select an already created column.

Example:
filter = rsi(14) > 70;
AddColumn("RSI", rsi(14));

This will create a column named 'RSI' that will contain the value returned by the formula rsi (14).

1.6. To update a watch list

To update a watch-list:
Select the 'watch-lists' tab in the 'Watch-list' form, select a 'watch-list', then click on the button 'Update'.

1.7. Watch lists grid colors

The grid line color is white when the symbol for this line is up to date. (The last quote date corresponds to the current date). The line color changes depending on the date difference between the last quote date and the current date.

1.8. Watch lists Settings

To update settings:
Select the 'Config' tab in the 'Watch list' form.

You can specify whether the watch list performs update automatically or not.

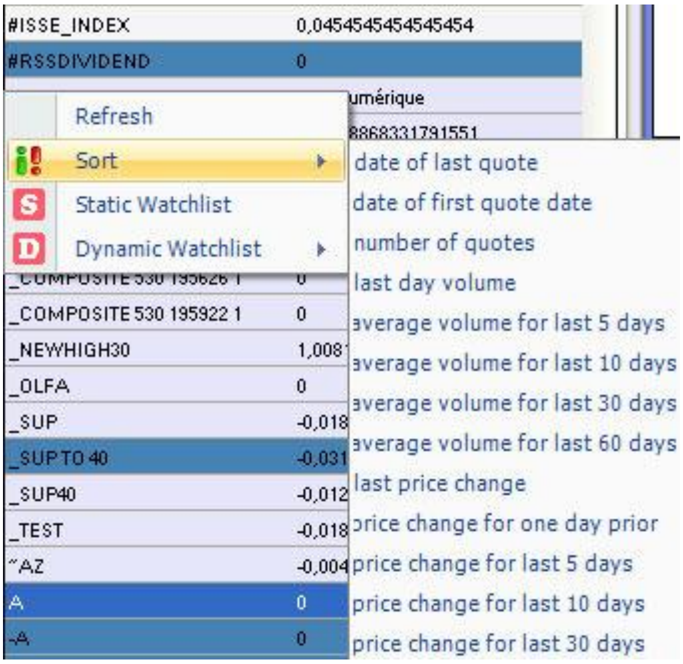
1.9. Switch between watch lists

To change the current watch list, right click on the grid. Select 'Static watch list' or 'Dynamic watch list' then select on a watch list.

1.10. Sort Column

The last grid's column contains Pre-Calculated values.
Right click on the grid then select 'Sort'; choose the item you want to use for this column.

Example of use: Select the item 'Distance from support' to see the distance between the last symbol quote and the support you have drawn for this symbol (if any).


#ISSE_INDEX 0,0454545454545454
#RSSDIVIDEND 0
umérique
8868331791551
Refresh
Sort
Static Watchlist
Dynamic Watchlist
_COMPOSITE 530 195922 1 0
_NEWHIGH30 1,008
_OLFA 0
_SUP -0,018
_SUP TO 40 -0,031
_SUP40 -0,012
_TEST -0,018
~AZ -0,004
A 0
-A 0
date of last quote
date of first quote date
number of quotes
last day volume
average volume for last 5 days
average volume for last 10 days
average volume for last 30 days
average volume for last 60 days
last price change
price change for one day prior
price change for last 5 days
price change for last 10 days
price change for last 30 days

[TOP]

1.2.6.5 Script Manager

1. Script Manager

1.1. Introduction

The script manager is a tool that let you write, update and manage your scripts.
Scripts are used to perform tasks and to control application behavior.
Scripts use CSharp or JScript.Net as programming languages.

1.2. Create a new Script

In 'Tools' menu, click on 'Script Editor'. Click then on 'File', then 'New'. Type a file name, and then click on 'OK'.
CTRL+SPACE shortcut shows you the available functions and variables.

Script Example:

```
var a = 2;

var i : int = 0;



var chart : Chart = Charts.GetSelectedChart();



for(i=0;i<1000;i++)

{

    chart.ScrollBarIndex = i;

    App.Sleep(30);

}
```

1.3. Open an existing script

To open an existing script in the 'Script Manager', click on 'File', then click on 'Open' and then select a file; a new tab will appears.

1.4. Execute a script

To compile and execute a script within the 'Script Manager', select the script you want to execute then click on 'Execute' in the menu bar

[TOP]

1.2.6.6 Widget Panel

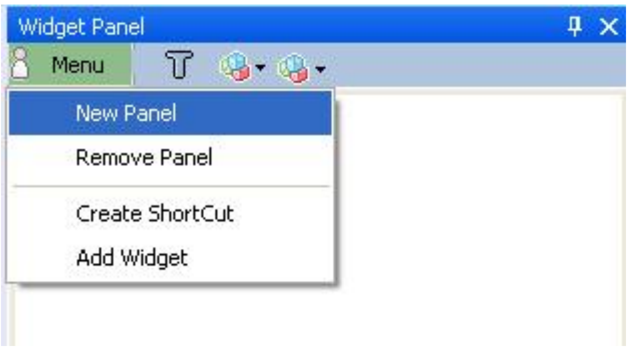
1. Widget Panel

1.1. Introduction

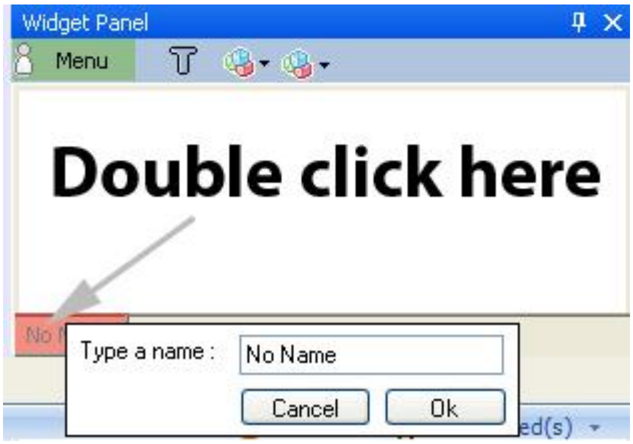
The widget panel is a container where you can add application object shortcuts or widgets. You can drag and drop objects like a pane formula, a trading system, a composite... to a widget panel.
You can also add widgets or .Net controls to the 'Widget Panel'.

1.2. Manage panels

Tools -> Widget Panel
In the 'Widget Panel' form, click on 'Menu' then click on 'Add panel' to create a new panel.



To rename a panel, double click on its name as shown below, then type a new name.

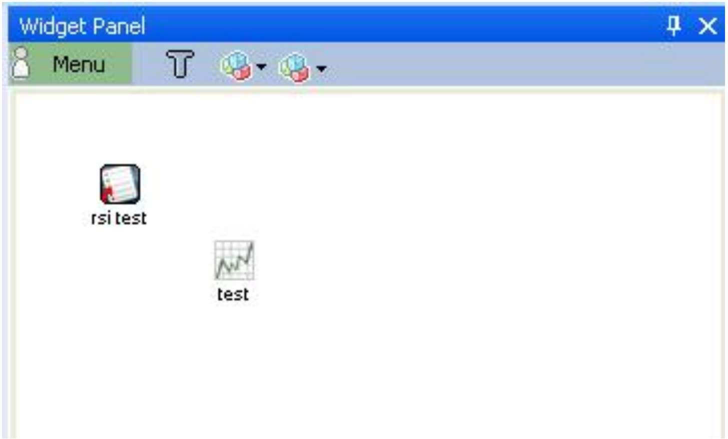


1.3. Create a shortcut

Click on 'Menu' then select 'Create Shortcut'

Select an item type
Check the check box if you want to get a shortcut for an existing item, otherwise you will create a shortcut for a new item
Follow the instructions, then click on 'Finish'

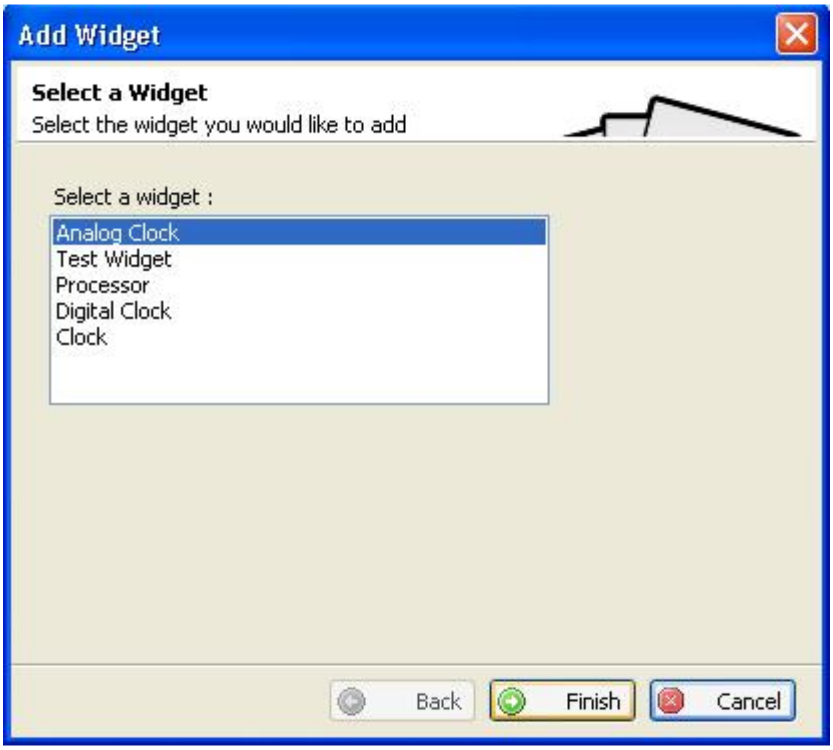
A shortcut will appears on your panel



1.4. Add a widget

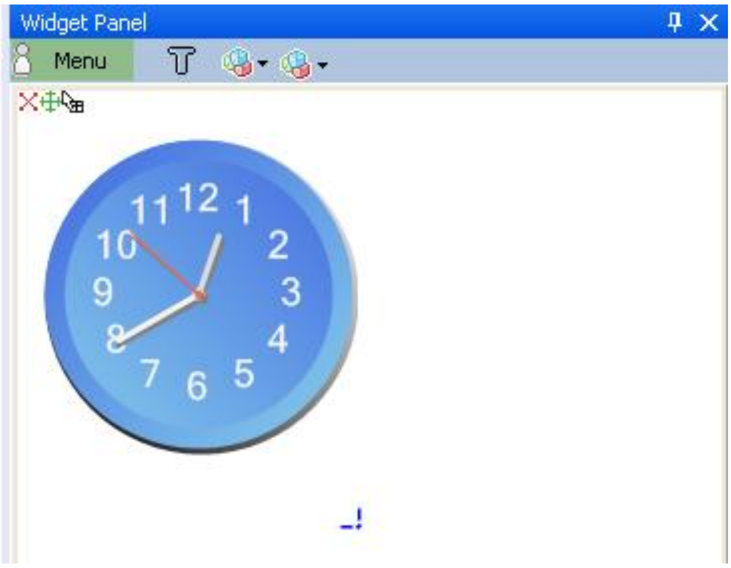
You can add a widget or .Net control.

Click on 'Menu' then select 'Add Widget'.
Select the control you would like to add then click on 'Finish'.


Add Widget

Select a Widget
Select the widget you would like to add

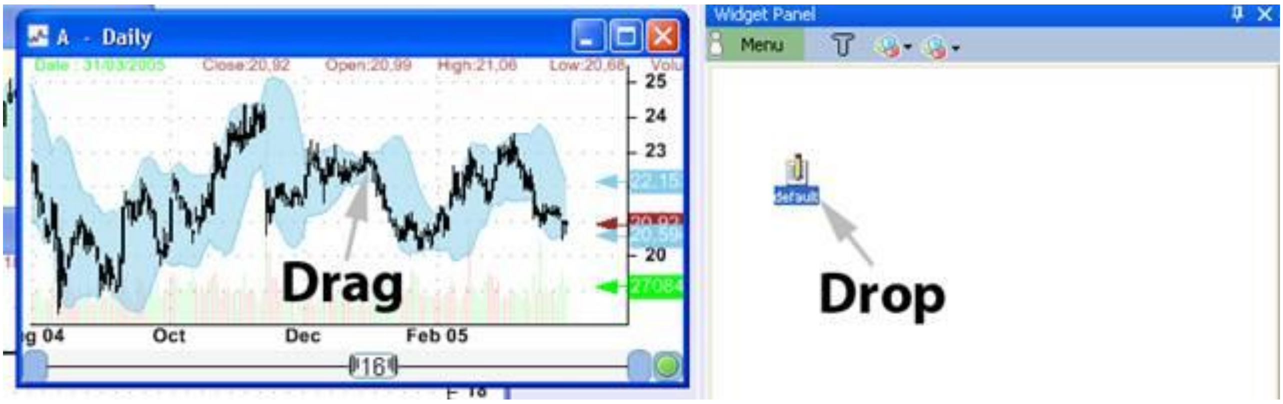
Select a widget :
- Analog Clock
- Test Widget
- Processor
- Digital Clock
- Clock



1.5. Drag & Drop

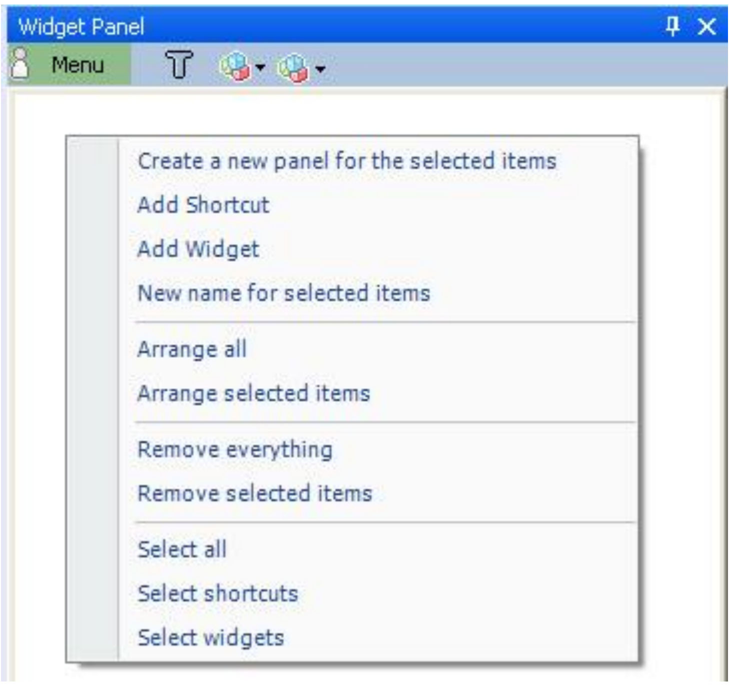
You can create a shortcut directly in your widget panel by dragging and dropping objects.

Example:
Create a chart then click on a time-series line or bar within that chart, drag & drop it to a widget panel, a shortcut will appears inside this panel.



1.6. Panel Menu

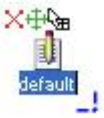
Right click on the panel to make the panel menu appears.



This menu contains the following options:

- Create new panel for the selected items
- Add shortcut
- Add widget
- Assign a new name for the selected items
- Arrange all the items
- Arrange the selected items
- Remove all the items
- Remove the selected items
- Select everything
- Select only shortcuts
- Select only widgets

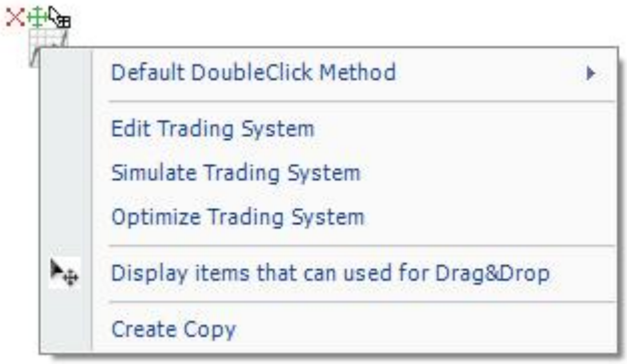
1.7. Shortcut and Controls Menu



Click on the Red Cross to remove this item.
Click on the arrow icon to drag this object, you can drag it to another panel or to a chart pane if the shortcut you are selecting is a formula.

Edit a shortcut name by clicking on its name.
Resize a shortcut by clicking on the blue line.
Right click on shortcut to make a menu appears.

Example: the menu of a trading system shortcut



[TOP]

1.2.6.7 Sharing Server

1. Sharing Server

1.1. Introduction

All the data used in the application except the databases, can be shared.
You can share and upload trading systems, list of rules, watch lists... and download others users items.
The sharing server is the place where you can search, comment, rate other users items.
The website we have developed is a combination of a sharing server and a social network.
You can communicate with others users, create and join groups, discuss about your favorite subjects, monitor which objects or items others users are downloading or uploading, download your favorite items, search for the most rated items...

1.2. Sharing Website

Sharing website benefits:

- See what others users are uploading and downloading
- Search for specific objects
- Create alerts and be informed when new objects that meet your criteria are uploaded
- Review, rate and report shared objects
- Send messages to others members
- Create and join groups
- Discuss with other users about your favorite topics.

1.3. Share an object

- Open the 'Share Items' form (*Tools -> sharing server*)
- Click on the 'Share' button at the top
- Select the item type
- Select an item, then click on the 'Share selected item' button, which is located on the right
- Read the guidelines
- Update the description if necessary
- Click on 'Share'

It is important that you follow the guidelines, because any shared object that does not meet the guidelines will be deleted.

1.4. Download an object

- Open the 'Share Items' form (*Tools -> sharing server*)
- Click on the 'Download' button at the top
- Select the item type
- Select the item that you want to download, then click on the 'Download' button, which is located on the right
- The 'Status' column shows updates about the downloading process
- Once the object is downloaded, the 'Save an item' dialog will appear
- Select a name and a category (not necessary) for this object, then click on 'Save'

Now, you can use this object in your application

1.5. Bookmarks

In the download panel and in the panel "My Lists", click on "Bookmarks" to see all the objects that you have bookmarked.

1.7. Sharing server website

QuantShare

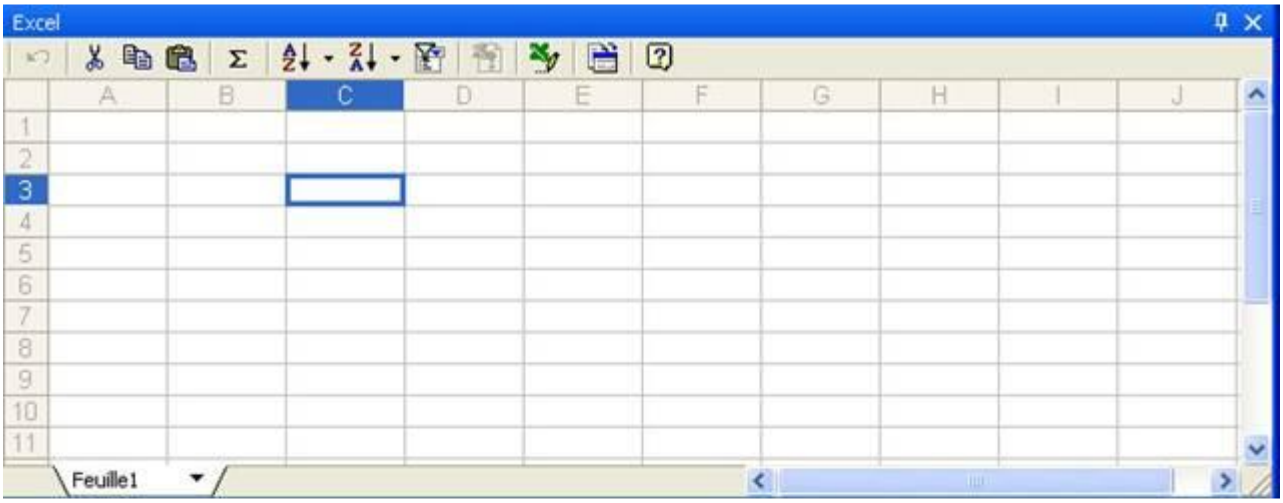
[TOP]

1.2.6.8 Divers

1. Divers

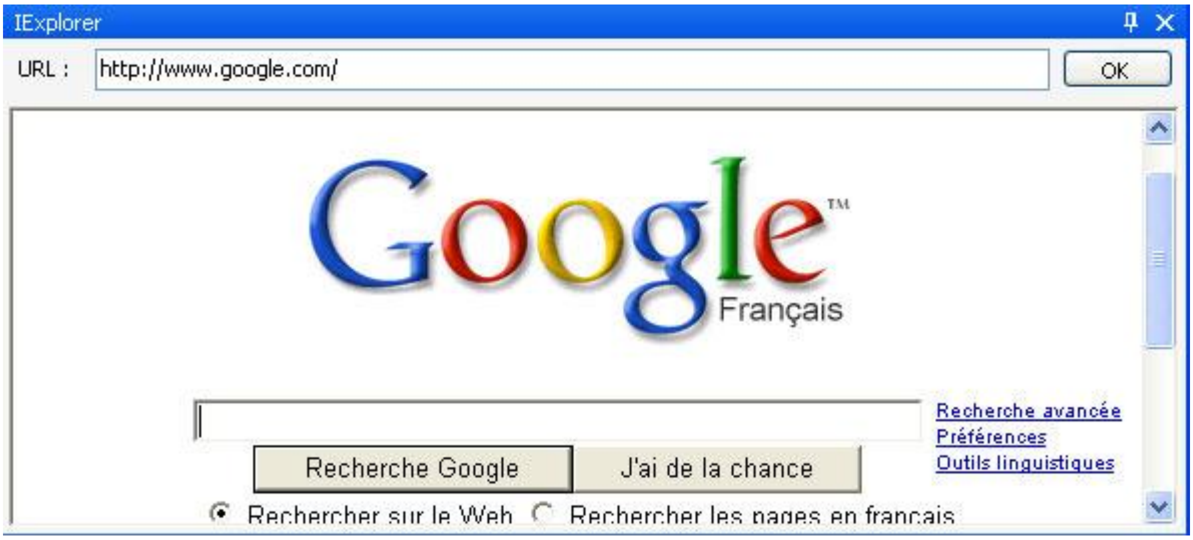
1.1. Excel

You can open an excel form within your application. To do so, click on 'Divers' menu then on 'Excel'.



1.2. Internet explorer

You can open an internet explorer form within your application. To do so, click on 'Divers' menu then on 'IExplorer'.


URL : http://www.google.com/

1.3. Output

Output form is a small form that displays text in a text-box.
You can add text to this form using the .Net function 'App.AddToOutput', or the QuantShare function: 'Output'.

1.4. Output list

Output form is a small form that displays text in a grid.
You can add text to this form using the .Net function 'App.AddToOutputList', or the QuantShare function: 'OutputList'.

[TOP]

1.2.6.9 Data

1.2.6.9.1 ASCII Importer

1. ASCII Importer

1.1. Introduction

The ASCII Importer is used to parse text or CSV files and include the parsed content into the application's databases. The ASCII imported has many settings that let you import any kind of data into quotes or custom databases.

N.B. the downloader plug-in use the ASCII Importer engine to parse the downloaded data.

1.2. Import Text or CSV data

To import data into the application's databases, click on 'Data' in the menu, then select 'ASCII Import'.

Once the import form opens, select one or many files by clicking on 'Pick files'.
If there are no reference to symbols in your files then you have to associate each file with a symbol name. To do so, just select a file in the list box, then type in a symbol name in the text box nearby the button 'Update symbol name', click on that button to associate a symbol to the file.

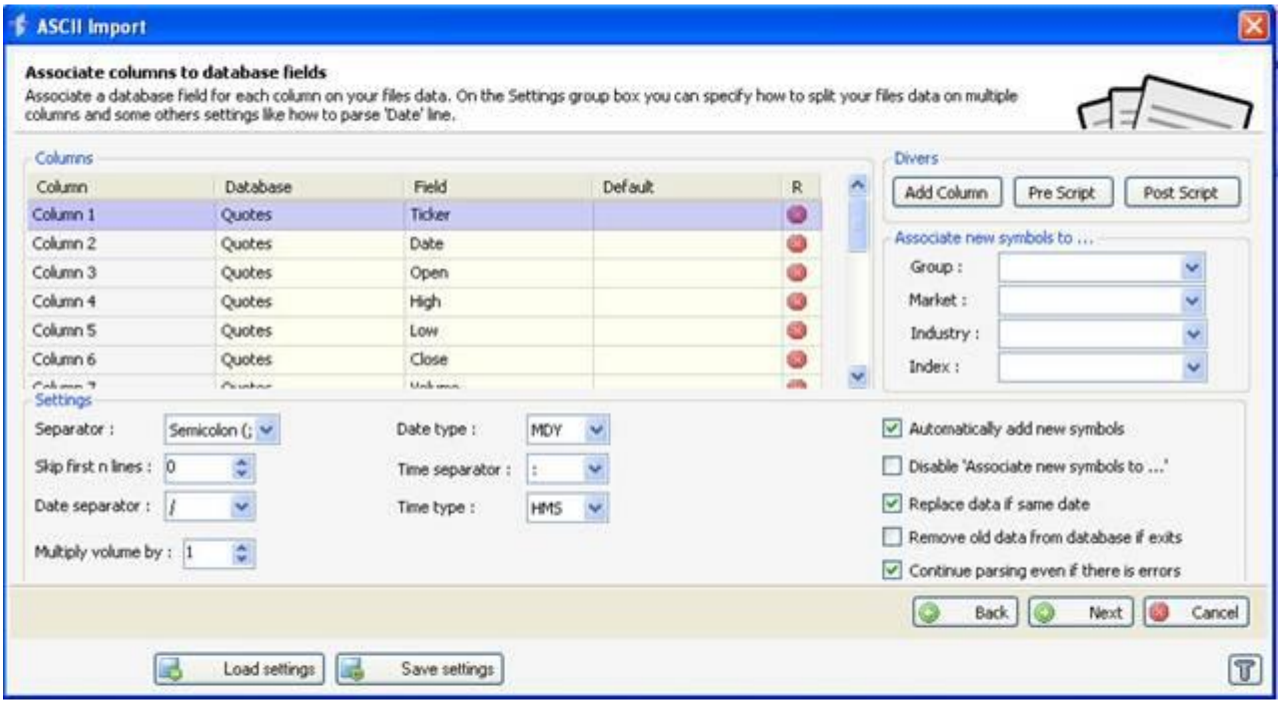
Note that you do not have to do that if there are references to symbols in your files content.

Click on 'Next' to go to the next screen. This screen let you define how to parse the content of your files, extract dates, symbols, execute script before or after parsing, associate symbols to groups, markets...

Let us take an example; we are going to show you how to parse the current lines:
A;01/12/08;12.2;13.3;11.5;12.1;1200
A;01/13/08;12.2;13.3;11.5;12.1;1200
A;01/14/08;12.2;13.3;11.5;12.1;1200

First, we have to define the separator, in this case its ';'.

We do not need to escape any lines.
We have to choose '/' as a separator for dates.
Since the date format is Month/Day/Year, we then choose MDY as 'Date type'.
Now we are going to create columns by clicking on the button 'Add Column'; we need to create seven columns.
For each column, we select 'Quotes' as database.
The first column field will be 'Symbol', the second one 'Date', the third 'Open', the forth 'High', the fifth 'Low', the seventh 'Close', the eighth 'Volume'.



Associate columns to database fields

Associate a database field for each column on your files data. On the Settings group box you can specify how to split your files data on multiple columns and some others settings like how to parse 'Date' line.

Columns

Column	Database	Field	Default	R
Column 1	Quotes	Ticker		☒
Column 2	Quotes	Date		☒
Column 3	Quotes	Open		☒
Column 4	Quotes	High		☒
Column 5	Quotes	Low		☒
Column 6	Quotes	Close		☒
Column 7	Quotes	Volume		☒

Associate new symbols to ...

Group :

Market :

Industry :

Index :

Settings

Separator : Semicolon (;

Skip first n lines : 0

Date separator : /

Multiply volume by : 1

Date type : MDY

Time separator : :

Time type : HMS

☑ Automatically add new symbols

□ Disable 'Associate new symbols to ...'

☑ Replace data if same date

□ Remove old data from database if exits

☑ Continue parsing even if there is errors

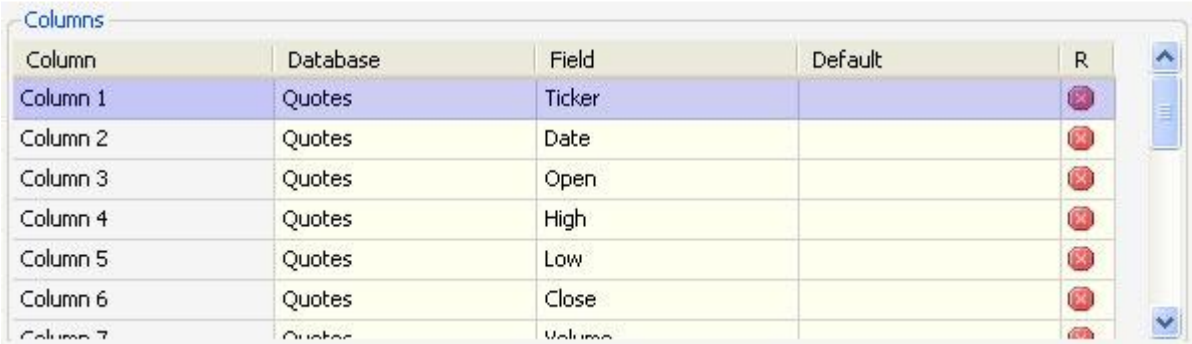
Click on 'Next' to go to the 'Input & Output' panel. You will get a sample of your files content as well as a sample of the parsed content.

You have to click on 'Database->Field' items to see what the output will look like.

Click again on 'Next', so that the application can parse your files. After parsing the data, you will get the number of errors if any, the number of symbols that will be added... If during the parsing the engine finds errors, go back to the previous panel and correct the errors; if there are no errors then click on 'Next' to start the importing process.

1.3. Columns

The columns are used to parse the content of the files. Look at the 'Import Text or CSV data' paragraph for an example.



Columns

Column	Database	Field	Default	R
Column 1	Quotes	Ticker		☒
Column 2	Quotes	Date		☒
Column 3	Quotes	Open		☒
Column 4	Quotes	High		☒
Column 5	Quotes	Low		☒
Column 6	Quotes	Close		☒
Column 7	Quotes	Volume		☒

Click on the Red Cross to remove a column.

You can associate a column to a quote or a custom database; double click on the cells that belong to the 'Database' column to see the available databases.

You can set 'Skip' as a field to skip the current column.

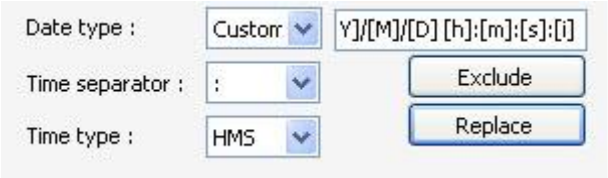
1.4. Import Settings

- Automatically add new symbols: Add the new symbols the parser finds.
- "Disable 'Associate new symbols to ...'": Make sure symbols will not be associated with any market, industry, group or index.
- "Replace data if same date": Override the existing data if data of the same date is found

- "Remove old data from database if exists": Before parsing, remove all the available content for the symbols that are going to be used in the import process.
- "Continue parsing even if there are errors": Continue the parsing process even if there are errors.
- "Text Qualifiers": Text between text qualifiers is treated as a single element.
  Example: (comma as separator and double quotes as text qualifiers)
  12,"12,5" -> element 1: (12) and element 2: (12,5).
  If we haven't specified text qualifiers, we would have: element1 : (12), element 2: ("12) and element 3: (5")

1.5. Advanced Date Format

In the 'ASCII Import' form, set 'Data type' to custom.



Date type : Custom Y]/[M]/[D] [h]:[m]:[s]:[i]

Time separator : :

Time type : HMS

The date format is defined as: [Y]/[M]/[D] [h]:[m]:[s]:[i]

[Y]: year with four digits, example: 2008.
{Y}: year with two digits, example: 08.
[M]: month
[D]: date
[h]: hour
[m]: minute
[s]: second
[i]: milliseconds

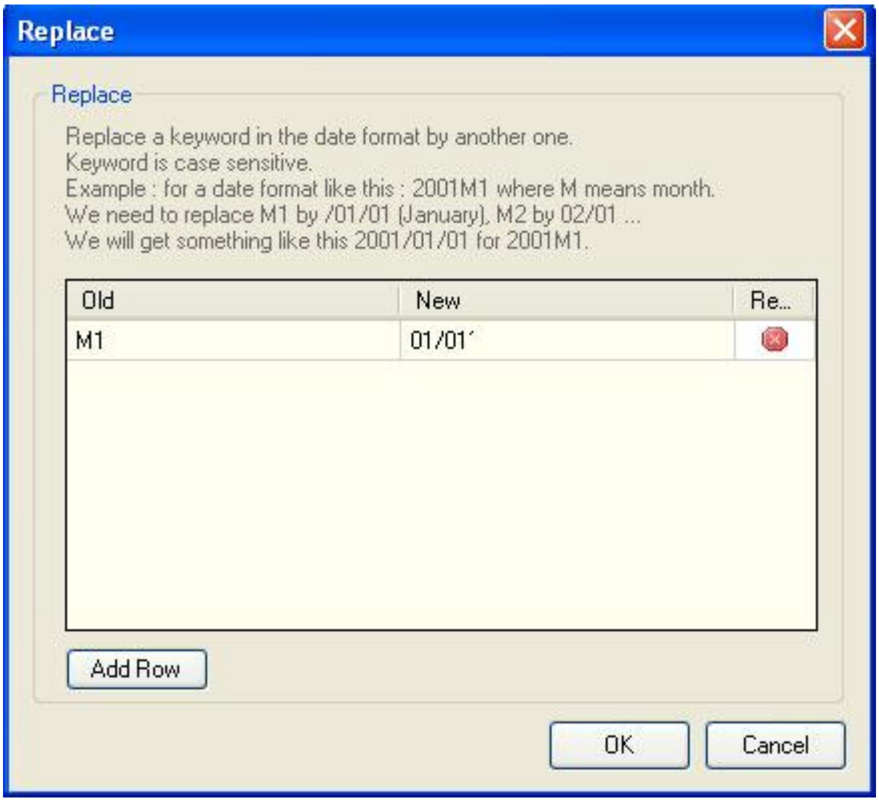
If the date you want to parse if for example: 12 01 2002 20:10, then set the date format to [D] [M] [Y] [h]:[m].

Exclude button:

Exclude keywords from a date.
Example: If your date is "Monday 12/01/2004", before parsing this date, you should get rid of the word 'Monday'. To do so, click on 'Exclude button' and insert the word 'Monday'.

Replace button:

Replace a keyword with another one.
Example: If your date is "2001M1" and M1 means January. You should click on replace button, and then click on 'Add row'. In that row, you have to type 'M1' under the 'old' column and type '/01/01' under the 'new column'. The date value will be "2001/01/01".



Replace a keyword in the date format by another one.
Keyword is case sensitive.
Example : for a date format like this : 2001M1 where M means month.
We need to replace M1 by /01/01 (January), M2 by 02/01 ...
We will get something like this 2001/01/01 for 2001M1.

Old	New	Re...
M1	01/01´	⊗

1.6. Pre-Parsing Script

Pre-Parsing Script is used to modify content before parsing begins.

Example:
Your content looks like this:
A;01/01/2008;A was upgraded by JP Morgan
B;01/01/2008;B was upgraded by JP Morgan
C;02/01/2008;C was downgraded by JP Morgan
D;01/01/2008;D was upgraded by JP Morgan

You want to create a column that contains data indicating whether the stock was downgraded or upgraded.

First, we should create four columns; set the databases and fields.

Now click on "Pre-Script" button.
At any moment you can press on 'CTRL+SPACE' to see the available variables.
In the Pre-Parsing script, we have one variable: "`Content`".

"Content" is of type "PreScript"
"Content" contains the content of each row or line. It also contains two functions:

AddRow: Add a new row or line. You must pass the elements to be added as parameters. The number of elements must be equal to the number of elements in the orginal content. There are three elements in our example.
Example of how to use "AddRow":

```
Content.AddRow("GOOG","01/01/2008","GOOG was upgraded then downgraded");
```

GetURLORFileName: Gets the URL of file name of the content we are going to parse.

Each row contains two variables: 'Data' and 'IsIgnoreLine'

"Data" is an array of string: Contains the parsed line data. In our example: (for line number 1) Data[0] contains 'A', Data[1] contains '01/01/2008'...
"IsIgnoreLine" is a Boolean: Specify whether to accept or ignore the current line or row.

The Pre-Script is executed for each file or document.

In our example, we want to check if the column number three (referenced by Data[2] -arrays start with the index zero-) contains the word 'Upgraded'.

Here is the script:

```
for(var i:int = 0;i<Content.Rows.Length;i++)
{
  if(Content.Rows[i].Data[2].Contains("Upgraded"))
  {
    Content.Rows[i].Data[3] = "1";
  }
  else
  {
    Content.Rows[i].Data[3] = "0";
  }
}
```

The fourth column value is '1' if the stock is upgraded, otherwise its value is '0'.
You can remove the 'else' statement in the script and set a default value of '0' in the fourth column.

1.7. Post-Parsing Script

Post-Parsing Script is used to modify parsed content.
Example of usage: Look at the parsed symbol quotes and correct bad ticks.

The Post-Parsing Script contains one variable: "Data"

"Data" is of type "PostScript"
"Data" contains the parsed content for each date.

Each row contains two variables:

Date: A date
Data: An array of objects (could be numeric 'double' or 'text') for the date referenced by the variable 'Date'
IsIgnoreLine: Specify whether to ignore the current row or not.

The Post-Script is executed for each symbol.

If your columns are defined like this:
Column 1: Symbol
Column 2: Date
Column 4: Open
Column 5: High
Column 6: Low
Column 7: Close
Column 8: Volume

In addition, your file content is:

```
A;13/12/2008;11;11;11;11;100
A;12/12/2008;12;12;12;12;200
```

Then variable 'Data' (of type PostScript) will contain two rows.
In the first row, the variable 'Date' will have '13/12/2008' as value and the variable 'Data' will be an array.
Data[0] = 11
Data[1] = 11
Data[2] = 11
Data[3] = 11
Data[4] = 100

[TOP]

1.2.6.9.2 Downloader

1. Downloader

The downloader plug-in downloads CSV, text, excel, compressed (ZIP), or RSS data from any website, parses it, transforms it, and then imports it to your databases.

N.B. the downloader plug-in uses the ASCII Importer engine to parse the downloaded data.

1.1. Introduction

Each download item is associated with one or sereral URLs, and each URL can contain any number of fields.

If for example you want to import stock quotes for the server 'www.example.com', and the complete URL to get stock quotes for the symbol "goog" is "www.example.com/quotes.php?id=goog".
In addition, you have to login using the following URL
"http://www.example.com/login.php?user=xx&pass=yy" before being able to download any quotes.

Given this example, your download item, should contains two URL:

1 – "http://www.example.com/login.php"
2 – "http://www.example.com/quotes.php"

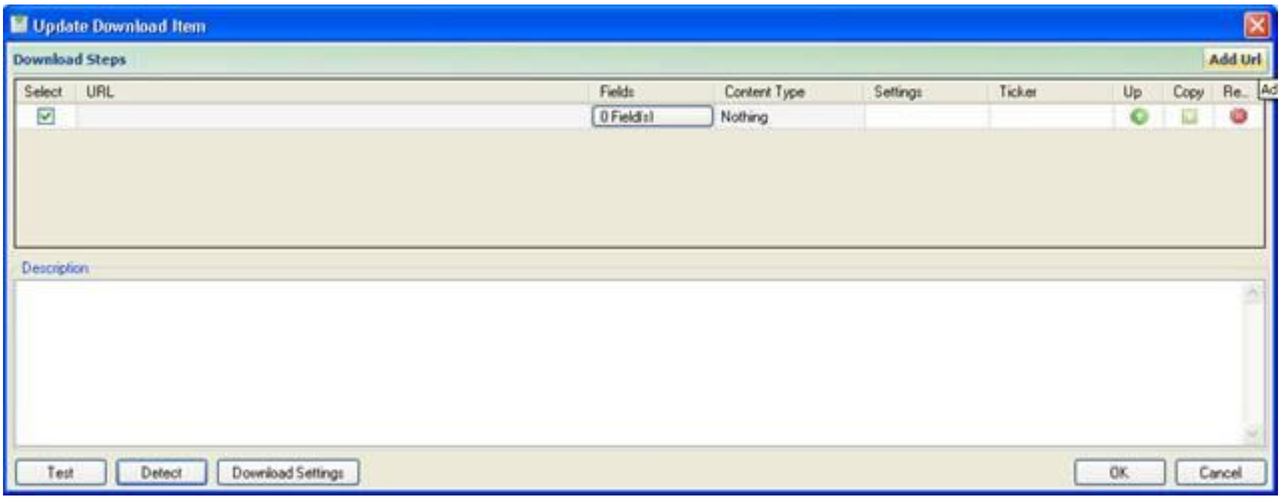
The first URL should contain two fields, one for the login and another one for the password.
The second URL should contain a field named 'id', you have to set this field's type to 'Symbol'.

1.2. Create a download item

Open the download manager by clicking on 'Download' in the menu bar, then 'Download Manager'.
Click on 'Add', in the download manager.

1.2.1. Download Steps

'Download Steps' are the URLs that the downloader must visit in order to download your content.
Click on 'Add URL' to add a new download step.



Update Download Item

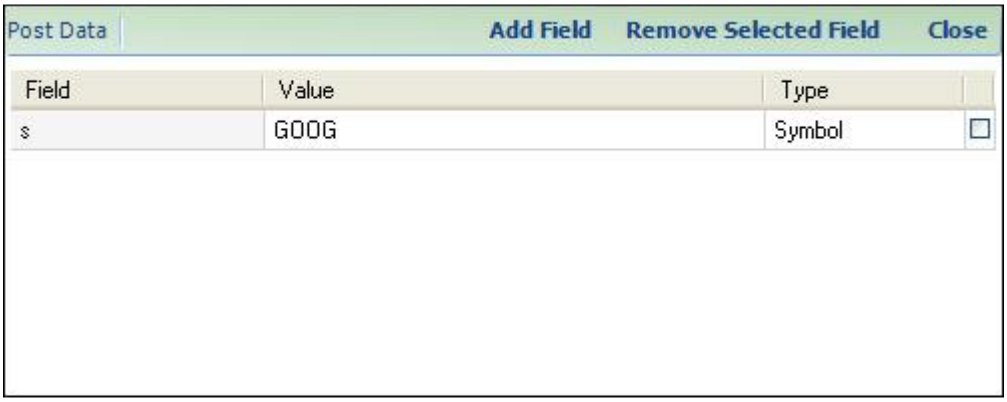
Download Steps

Select	URL	Fields	Content Type	Settings	Ticker	Up	Copy	Re..
☑		0 Field(s)	Nothing			●	□	●

Description

1.2.2. Fields

You can associate one or more fields to each 'Download Step' or URL.
First, add a new URL, and then click on the button 'Fields' under the 'Fields' column. Fields are used as parameters.



Field	Value	Type	
s	GOOG	Symbol	□

To add new fields, click on 'Add Field'.
To remove a field, select one, and then click on 'Remove Selected Field'.
To close the fields form, click on 'Close'.

When adding a field, you should specify four things:

- Field name: The name that will be used in the URL
- Field value: The value that will be used in testing and in static fields like the login and password fields; the value of a dynamic field is automatically updated in the downloading process.
- Field Type: The type associated with the field.
- Field visibility: Indicate whether to display the field as a URL parameter or not. Check the box to hide the field. You still can use the fields data (please refer to how to use brackets).

Example:

URL: "www.example.com/test.php?a=2"
Field Name: b
Field Value: test
Field Type: Custom

In the testing process, the software will use this URL: www.example.com/test.php?a=2&b=test.
In the downloading process, if you put the following values (EX1 and EX2) in the text box inside the tab associated with the field "b":

The downloader will use theses URLs to download your content:
"www.example.com/test.php?a=2&b=EX1"
"www.example.com/test.php?a=2&b=EX2"

How to use brackets:
In order to use a field inside a URL you should insert the field name inside brackets.
Let us take the last example: "www.example.com".

You can display the value "2" that corresponds to the field "a", using the following text: [a]
Example: "www.example.com/test_[a].php".

Keywords:
[DATE]: displays the date
[SYMBOL]: displays the symbol
[field_name]: displays any field value

1.2.3. Ordering URLs

Open the 'Update Download Item' form.
Click on the icon within the 'Up' column.

1.2.4. CSV, Excel, RSS, ZIP

After you have added the URL, specified the fields, and depending on the content you are parsing, click on the cell within the 'Parser' column and select the appropriate content type.
Click on 'Parser' to open the 'Parser' form.
This form will upload some content to help you fill in the appropriate parsing settings.
Look at the 'ASCII Import' plug-in for more information.

Excel: Download excel file and parse the content of every sheet.
RSS: Download RSS feed then transform it to CSV.
Zip: Download a compressed archive and parse every file included inside the archive.

1.2.5. Settings

Click on the "Settings" cell to open the settings form.
Within this form you can specify:
Whether to submit the fields data with a POST or GET method.
Whether to execute the URL-Script once or for each combination of fields.
For more information on the URL-Script, please visit the appropriate section.

The URL-Script is executed before downloading any data. This script lets you define dynamic URLs. It is executed for each combination of fields, which means that if for example you are using 10 symbols and a custom field with 2 custom data, this script will be executed 20 times, once for each "Symbol - Custom data" combination. If you check the above element (execute the URL-Script once), then the script will be executed only one time. You will have to define the URLs inside this script.

1.2.6 Test

In the 'Update Download Item' form, there is a button named 'Test'.
Click on this button to see how the application will browse your URLs to import your content.
This tool is designed to help you find and correct possible problems.

1.2.7. Detect

In the 'Update Download Item' form, there is a button named 'Detect'.
Click on this button to open a browser page. Visit your content pages URLs to look for all the form fields names.
This tool is designed to help you create your download item.

1.3. Download Settings

1.3.3. Settings

- **Maximum number of download threads to allow:** specify how many concurrent threads to use during the downloading process.
- **This item should download data every:** You specify here a number of days for your download item, and the application will alert you each time a download item needs to be run. A column named 'Need a run', in the 'Download Manager' form, tells you if the download item needs a run or not.
- **Number of seconds to wait between requests**
- **Database to use for the 'last symbol date':** Use this option to download only missing quotes or data. Select a database so the downloader can grab the last date from the symbol data in the specified database.

1.3.4. Proxies

Specify a list of proxy URLs to use when downloading data.
Each line represents a proxy URL.

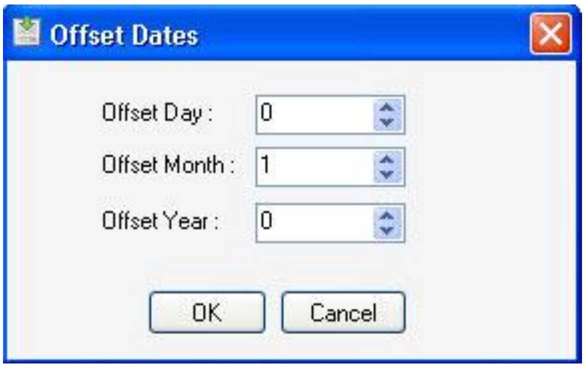
1.3.5. Symbol translation table

Translate symbols before downloading data.
Change the symbol name in the URL that will be used to download data.

Each line represents the source and destination symbol pair.

1.3.6. Offset dates

The "offset dates" button let you offset dates components.
If for example yahoo wants the month component to vary from 0 to 11, 0 means January and 11 means December, then in the 'Offset Dates', set the 'Offset Month' numeric box to one.

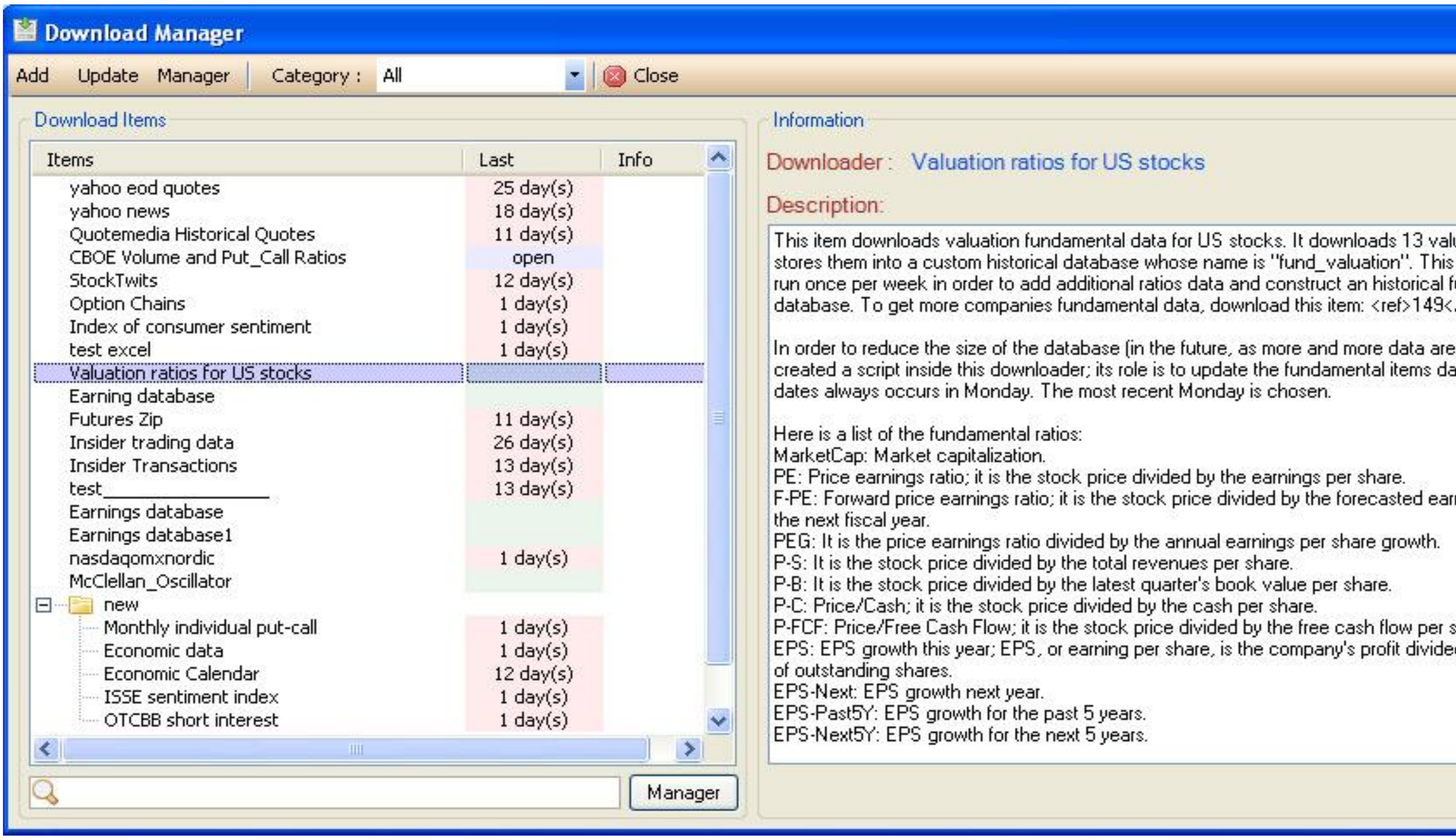


Offset Day : 0
Offset Month : 1
Offset Year : 0

1.4. Using the downloader

To open the downloader form, select "Download -> Download Manager".
Select a download item then click on "Open".
Click on the button "Start" to begin the downloading process.
The progress column shows you the download progression and each time a download is complete, the corresponding row is unchecked.

In the "download items" grid, the "Last" column indicates the number of days since the last execution of the corresponding item.



Items	Last	Info
yahoo eod quotes	25 day(s)	
yahoo news	18 day(s)	
Quotemedia Historical Quotes	11 day(s)	
CBOE Volume and Put_Call Ratios	open	
StockTwits	12 day(s)	
Option Chains	1 day(s)	
Index of consumer sentiment	1 day(s)	
test excel	1 day(s)	
Valuation ratios for US stocks		
Earning database		
Futures Zip	11 day(s)	
Insider trading data	26 day(s)	
Insider Transactions	13 day(s)	
test_______________	13 day(s)	
Earnings database		
Earnings database1		
nasdaqomxnordic	1 day(s)	
McClellan_Oscillator		
new		
Monthly individual put-call	1 day(s)	
Economic data	1 day(s)	
Economic Calendar	12 day(s)	
ISSE sentiment index	1 day(s)	
OTCBB short interest	1 day(s)	

Downloader : Valuation ratios for US stocks

Description:

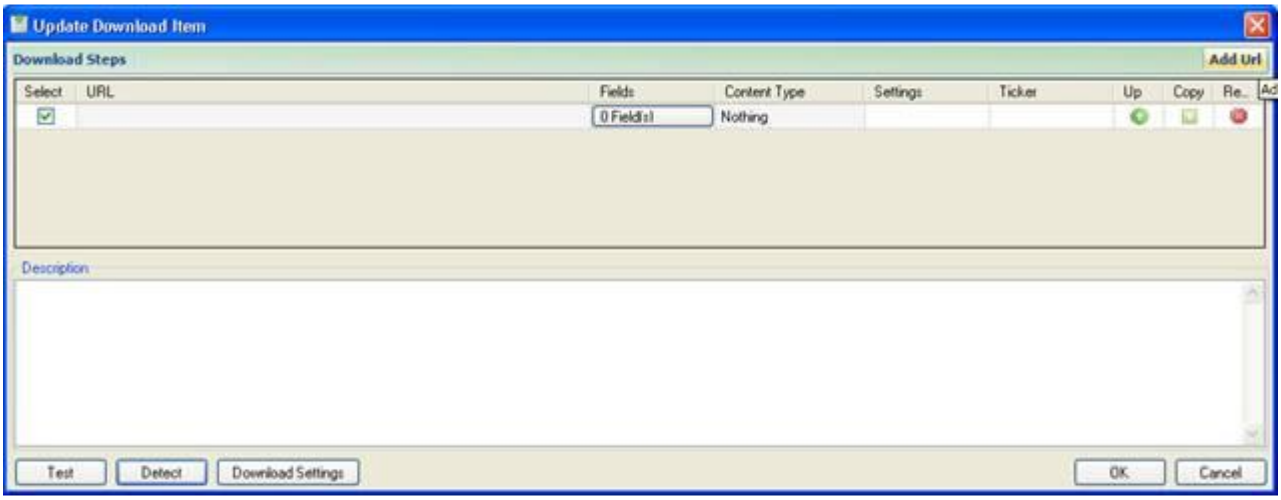
This item downloads valuation fundamental data for US stocks. It downloads 13 valu
stores them into a custom historical database whose name is "fund_valuation". This
run once per week in order to add additional ratios data and construct an historical f
database. To get more companies fundamental data, download this item: <ref>149<.

In order to reduce the size of the database (in the future, as more and more data are
created a script inside this downloader; its role is to update the fundamental items da
dates always occurs in Monday. The most recent Monday is chosen.

Here is a list of the fundamental ratios:
MarketCap: Market capitalization.
PE: Price earnings ratio; it is the stock price divided by the earnings per share.
F-PE: Forward price earnings ratio; it is the stock price divided by the forecasted ear
the next fiscal year.
PEG: It is the price earnings ratio divided by the annual earnings per share growth.
P-S: It is the stock price divided by the total revenues per share.
P-B: It is the stock price divided by the latest quarter's book value per share.
P-C: Price/Cash; it is the stock price divided by the cash per share.
P-FCF: Price/Free Cash Flow; it is the stock price divided by the free cash flow per s
EPS: EPS growth this year; EPS, or earning per share, is the company's profit divide
of outstanding shares.
EPS-Next: EPS growth next year.
EPS-Past5Y: EPS growth for the past 5 years.
EPS-Next5Y: EPS growth for the next 5 years.

1.4.1. Symbols

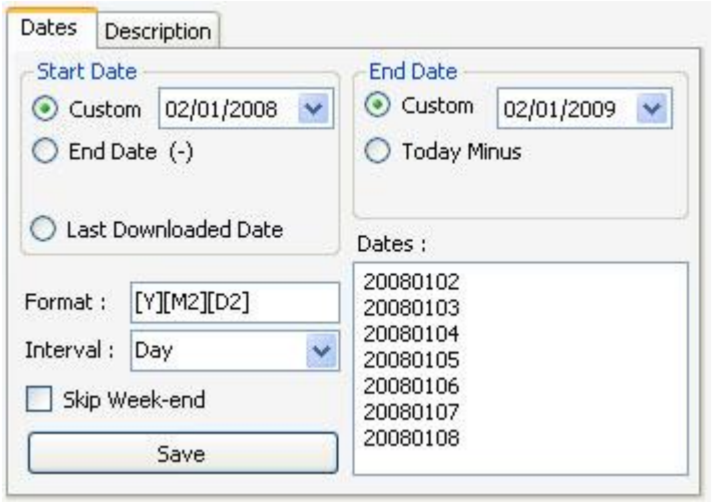
If you have specified a symbol field in the 'Update Download Item' form, then a 'Symbols' tab will appear. Select the symbols you would like to download.



Select	URL	Fields	Content Type	Settings	Ticker	Up	Copy	Re..
☑		0 Field(s)	Nothing			●	□	●

1.4.2. Dates

If you have specified a Date field in the 'Update Download Item' form, then the 'Dates' tab will appears.



Start Date
● Custom 02/01/2008
○ End Date (-)
○ Last Downloaded Date

End Date
● Custom 02/01/2009
○ Today Minus

Format : [Y][M2][D2]
Interval : Day
□ Skip Week-end

Dates :
20080102
20080103
20080104
20080105
20080106
20080107
20080108

In this tab, select the start date, the end date, the format, the interval, then click on 'Save'.

The 'Dates' text box, will show you the dates that are going to be used in the downloading.

Format text box:

[Y]: year, example: 2008
[Y2]: year, example: 08
[M]: month, example: 01 or 10
[M2]: month, example: 1 (without the 0) or 10
[D]: day, example: 01 or 22
[D2]: day, example: 1 (without the 0) or 22

In 'Update Download Item' form, you can also specify dates component within the URL.

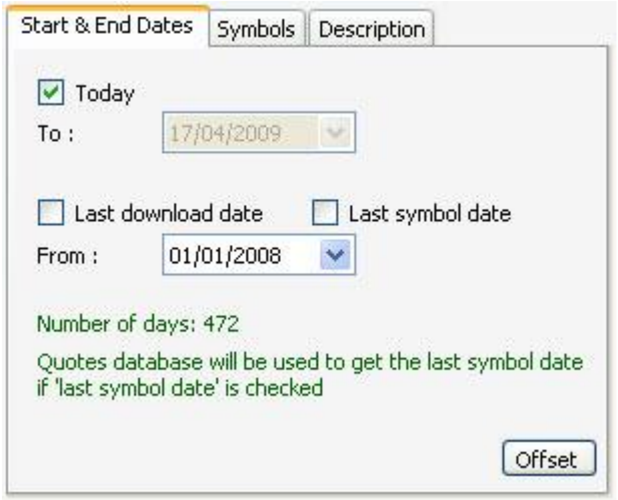
Example for yahoo:
http://ichart.finance.yahoo.com/table.csv?a=[2M2]&b=[2D2]&c=[2Y]&d=[M2]&e=[D2]&f=[Y]&g=d&ignore=.csv

Dates format are a little bit different here:
[Y], [Y2], [M], [M2], [D], [D2] refers to first date (FROM), while [2Y], [2Y2], [2M], [2M2], [2D], [2D2] refers to second date (TO).

1.4.3. Start and end dates

If you have specified date components within URLs, a 'Start & End Dates' tab will appears in the 'Download Data' form.

Select 'Last symbol data' to download only missing quotes or data.
The database that will be used to get the last downloaded date for a particular symbol can be set in the 'Download Settings' form.



☑ Today

To : 17/04/2009

□ Last download date □ Last symbol date

From : 01/01/2008

Number of days: 472

Quotes database will be used to get the last symbol date if 'last symbol date' is checked

1.4.4. Custom fields

If you have specified a custom field in the 'Update Download Item' form, then a tab containing a text box will appear. Each line inside this text box corresponds to a value.

1.5. Downloader Scripts

1.5.1. URL-Script

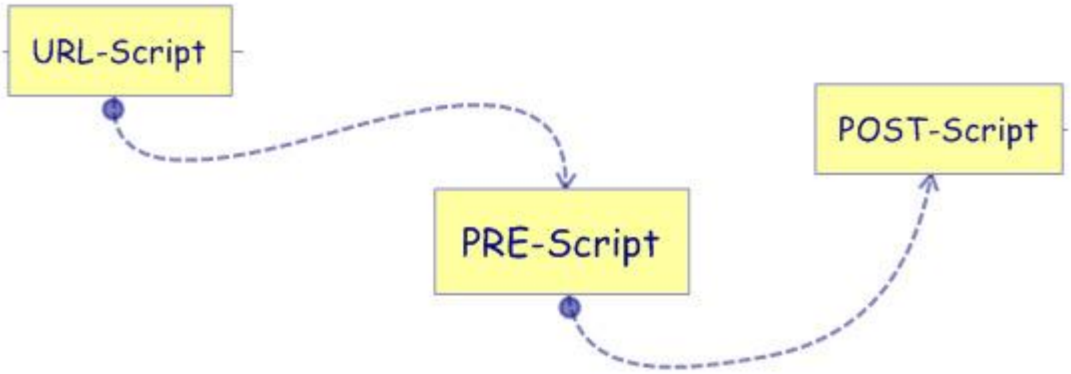
If you need to dynamically specify the URL to download then you have to use the URL-Script to programmatically specify what URLs or URLs paths to use.
A URL path is a sequence of URLs. It is used if for example before loading a URL, the website requires that you load some other pages, login...
A URL is added using the following function:
Functions.AddURL

A URL path is added using the following functions:
Functions.CreateURLPath: creates an URL path.
AddURL: adds an URL to an URL path.
Functions.AddURLPath: adds an URL path.

The (*Functions.Net*) class contains different methods to download, extract and parse HTML documents.

1.5.2. Summary

The engine will look for URLs specified in the URL-Script; it downloads them and split the content using the parser settings. It then passes the split content to the Pre-Script. This script allows you to modify the provided data. Finally the content is parsed and before added to the quotes or custom databases, it is passed to the Post-Script.

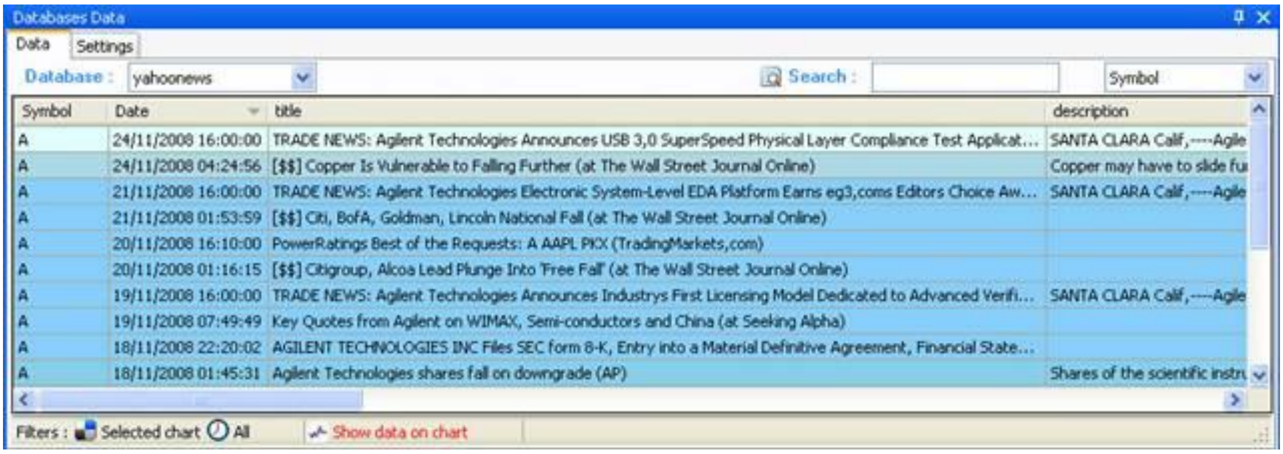

URL-Script
PRE-Script
POST-Script

1.2.6.9.3 Data Viewer

1. Data Viewer

1.1. Introduction

The data viewer lets you display database data inside a grid.

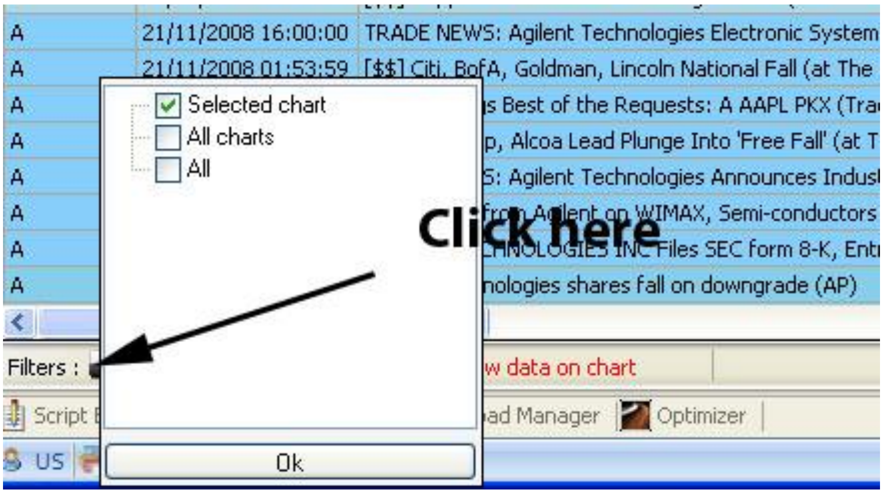


Database : yahoonews Search : Symbol

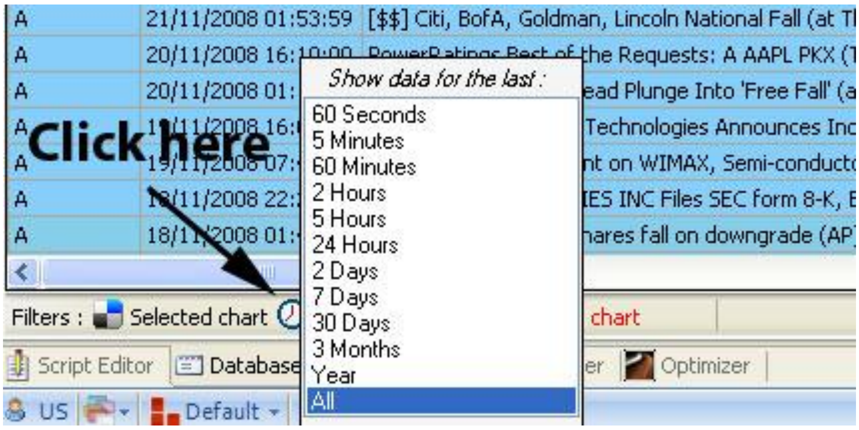
Symbol	Date	title	description
A	24/11/2008 16:00:00	TRADE NEWS: Agilent Technologies Announces USB 3,0 SuperSpeed Physical Layer Compliance Test Applicat...	SANTA CLARA Calif,---Agile
A	24/11/2008 04:24:56	[$$] Copper Is Vulnerable to Falling Further (at The Wall Street Journal Online)	Copper may have to slide fu
A	21/11/2008 16:00:00	TRADE NEWS: Agilent Technologies Electronic System-Level EDA Platform Earns eg3,coms Editors Choice Aw...	SANTA CLARA Calif,---Agile
A	21/11/2008 01:53:59	[$$] Citi, BofA, Goldman, Lincoln National Fall (at The Wall Street Journal Online)	
A	20/11/2008 16:10:00	PowerRatings Best of the Requests: A AAPL PKX (TradingMarkets,com)	
A	20/11/2008 01:16:15	[$$] Citigroup, Alcoa Lead Plunge Into 'Free Fall' (at The Wall Street Journal Online)	
A	19/11/2008 16:00:00	TRADE NEWS: Agilent Technologies Announces Industrys First Licensing Model Dedicated to Advanced Verifi...	SANTA CLARA Calif,---Agile
A	19/11/2008 07:49:49	Key Quotes from Agilent on WIMAX, Semi-conductors and China (at Seeking Alpha)	
A	18/11/2008 22:20:02	AGILENT TECHNOLOGIES INC Files SEC form 8-K, Entry into a Material Definitive Agreement, Financial State...	
A	18/11/2008 01:45:31	Agilent Technologies shares fall on downgrade (AP)	Shares of the scientific instru

1.2. Show database data

To open the 'database data' form; select the 'Tools' menu then click on 'Databases Data'.
To display database data, you need to select a database by selecting one in the 'Database' combo box, then click on the icon next to Filters and select for which symbols you want to display data. If you select 'Selected chart', then the grid content will be automatically updated each time the select chart symbol changes.



Click on the watch icon, depending on the period of time you select, the plug-in will only display data that meets the time constraint.



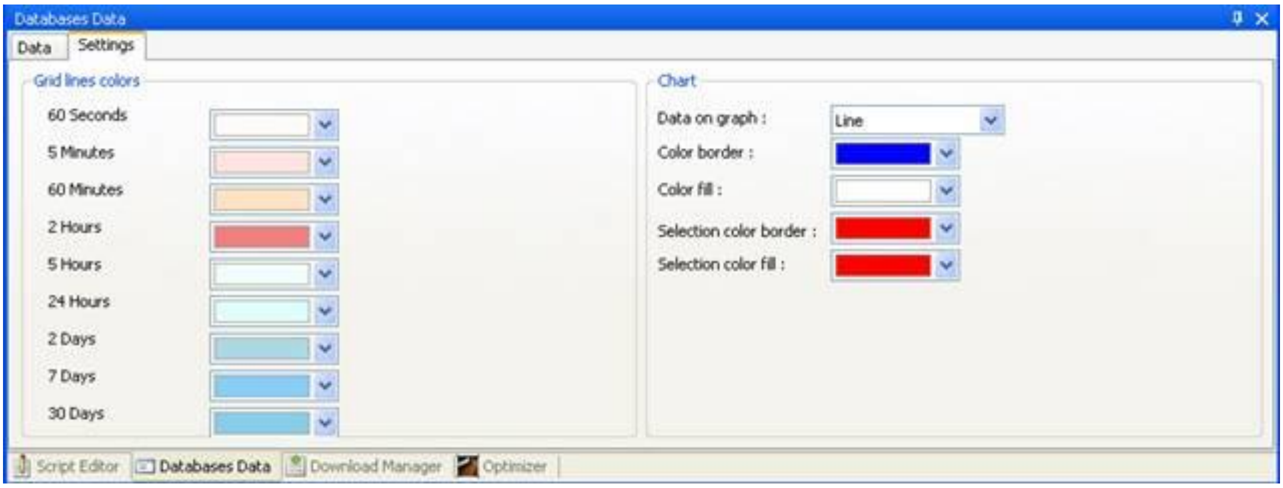
A | 21/11/2008 01:53:59 | [$$] Citi, BofA, Goldman, Lincoln National Fall (at T
A | 20/11/2008 16:10:00 | PowerRatings Best of the Requests: A AAPL PKX (1
A | 20/11/2008 01: | ead Plunge Into 'Free Fall' (a
A | 19/11/2008 16: | Technologies Announces Inc
A | 19/11/2008 07: | nt on WIMAX, Semi-conduct
A | 18/11/2008 22: | IES INC Files SEC form 8-K, E
A | 18/11/2008 01: | hares fall on downgrade (AP

1.3. Filter database lines

You can filter database data, by typing keywords in the 'Search' text box and by selecting a database field.

1.4. Data Viewer Settings

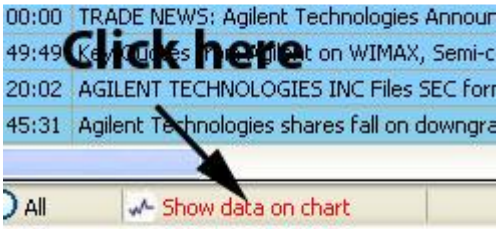
Each grid line has a different color depending on the age of the data. You can choose the colors by selecting the 'Settings' tab in the 'Data Viewer' form.



Data on graph : Line

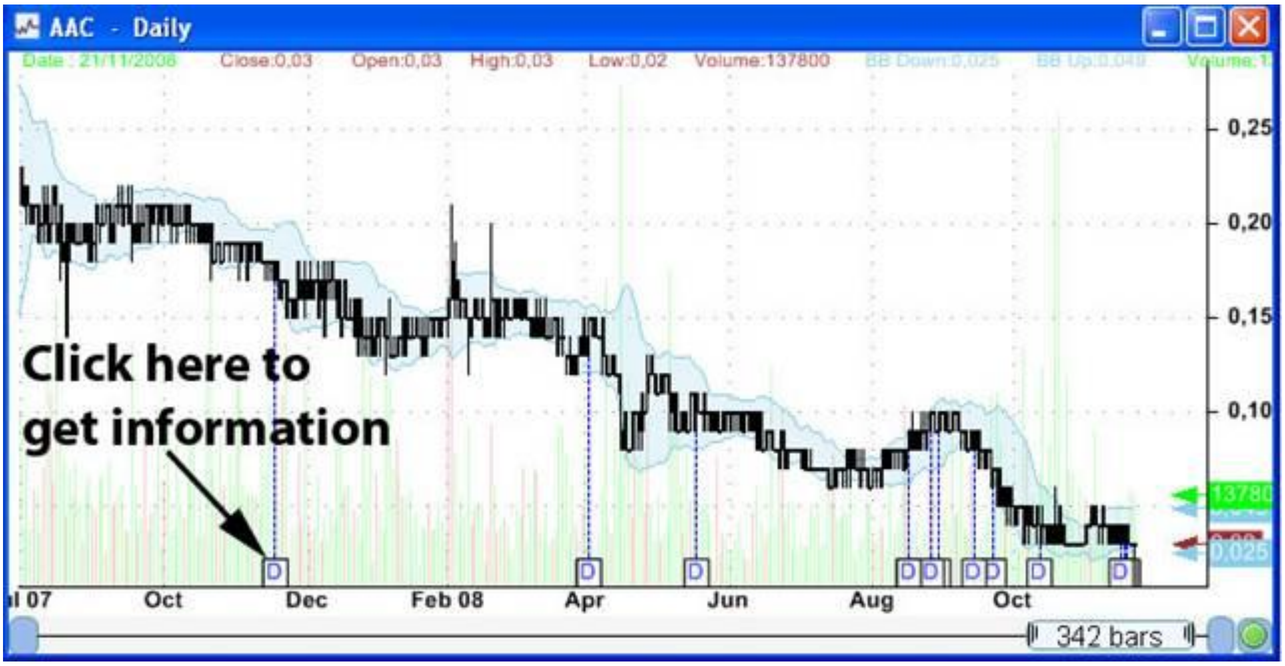
1.5. Plot data on the chart

To plot data on a chart, click on 'Show data on chart' in the 'Data Viewer' form.



00:00 | TRADE NEWS: Agilent Technologies Annour
49:49 | Keynote Speaker at ... on WIMAX, Semi-c
20:02 | AGILENT TECHNOLOGIES INC Files SEC forr
45:31 | Agilent Technologies shares fall on downgra

Click on the chart rectangle or circle to display information.
You can control how to display data on the chart in the 'Settings' tab panel.


AAC - Daily
Date : 21/11/2008
Close:0,03
Open:0,03
High:0,03
Low:0,02
Volume:137800
BB Down:0,025
BB Up:0,049
Volume:1
0,25
0,20
0,15
0,10
13780
0,025
Click here to get information
D
l 07
Oct
Dec
Feb 08
Apr
Jun
Aug
Oct
342 bars

[TOP]

1.2.6.10 Analysis

1.2.6.10.1 Rules Manager

1. Rules Manager

1.1. Introduction

The Rules manager allows you to manage and create easily and quickly as many rules as you want.

A rule is a formula that can be used in charting, simulation...
A list of rules is a list that contains one or many rules.
A rule can have one or several iterations.

You can create undefined variables within a rules and specify the start, the end and the increment value of these variables.
This process makes it easy to create hundreds of rules.

List of rules can be used in many situations.

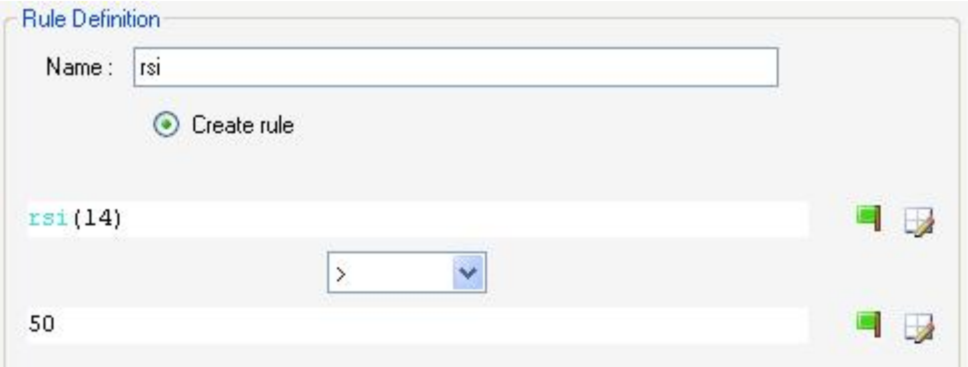
Examples:
You can back-test rules to see how each rule performs.
You can use the optimizer, and run a genetic algorithm process on a list of rules to see which combination of rules performs best.
When creating a trading system, you can define already created rules as buy, sell, and short or cover entries.

1.2. Create a rule

Open the rule manager (*Analysis -> Rules Manager*)
Click on 'Add' to create a list of rules
Select the list of rules you have just created.

Here is how to create a rule.
If you rule is something like x > y or x = y, you just have to type the first component in the first input box, and the second one in the second input box.

Example: rsi (14) > 50



Rule Definition

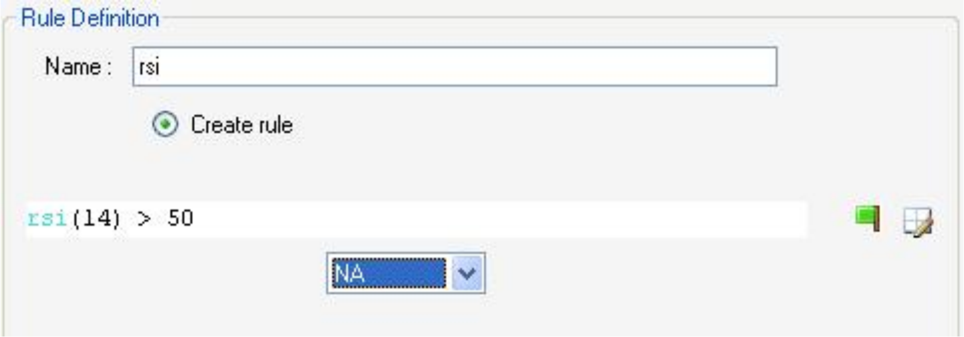
Name : rsi

● Create rule

```
rsi(14)
>
50
```

A red flag means that there is an error in the formula, if everything is ok then the red drag will turn into a green one.

In the combo box you can select NA, and then type directly the formula in the first input

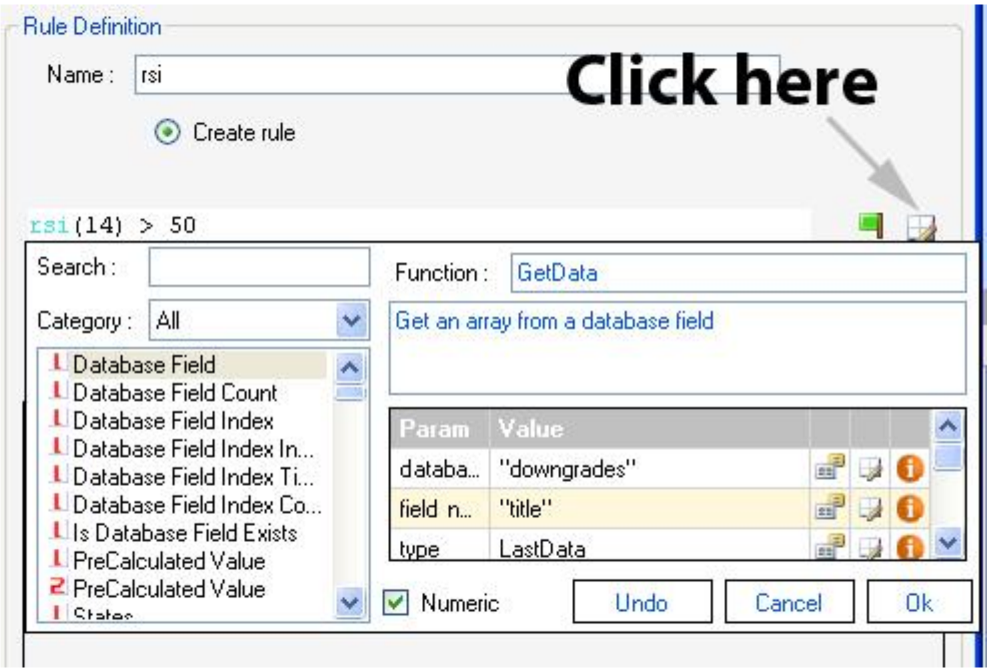


Name : rsi

○ Create rule

```
rsi(14) > 50
```

When you click on the icon next to the red flag, a small form will appear.
This form contains all functions definitions and parameters, use it to select the function you would like to include in your formula then click on OK.



Name : rsi

```
rsi(14) > 50
```

Category : All

Function : GetData

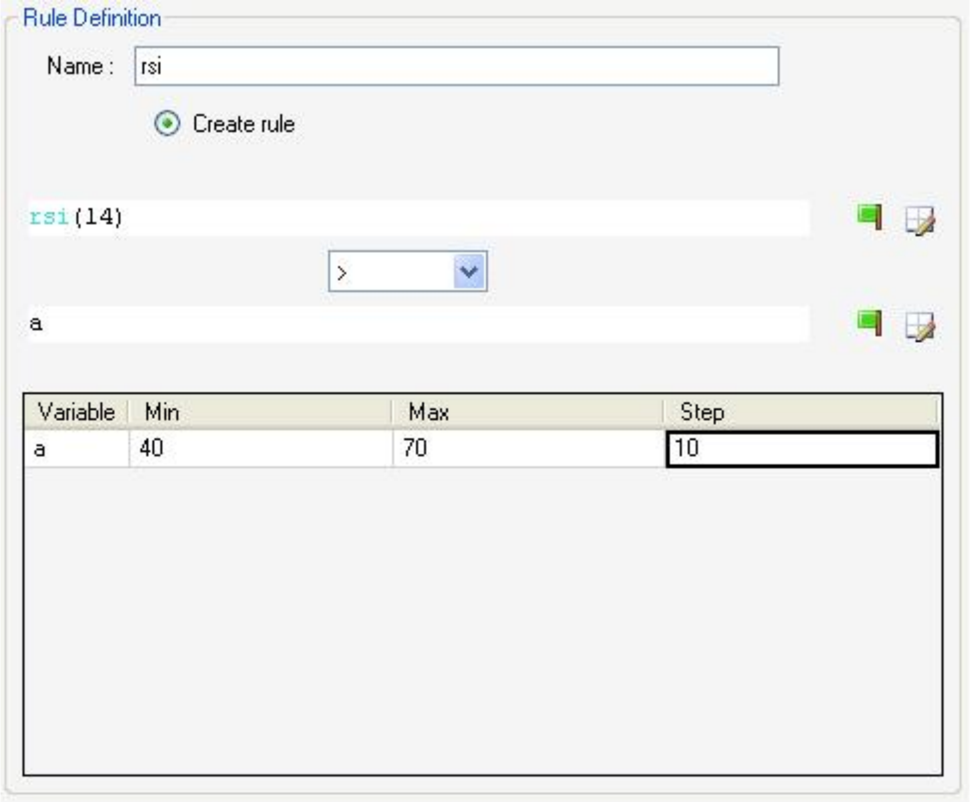
Get an array from a database field

Param	Value
databa...	"downgrades"
field n...	"title"
type	LastData

☑ Numeric

The grid below the inputs boxes is used to transform one rule into several ones.

Example: instead of specifying "rsi (14) > 50", you can type "rsi (14) > a", a line in the grid will appear.



Name : rsi

● Create rule

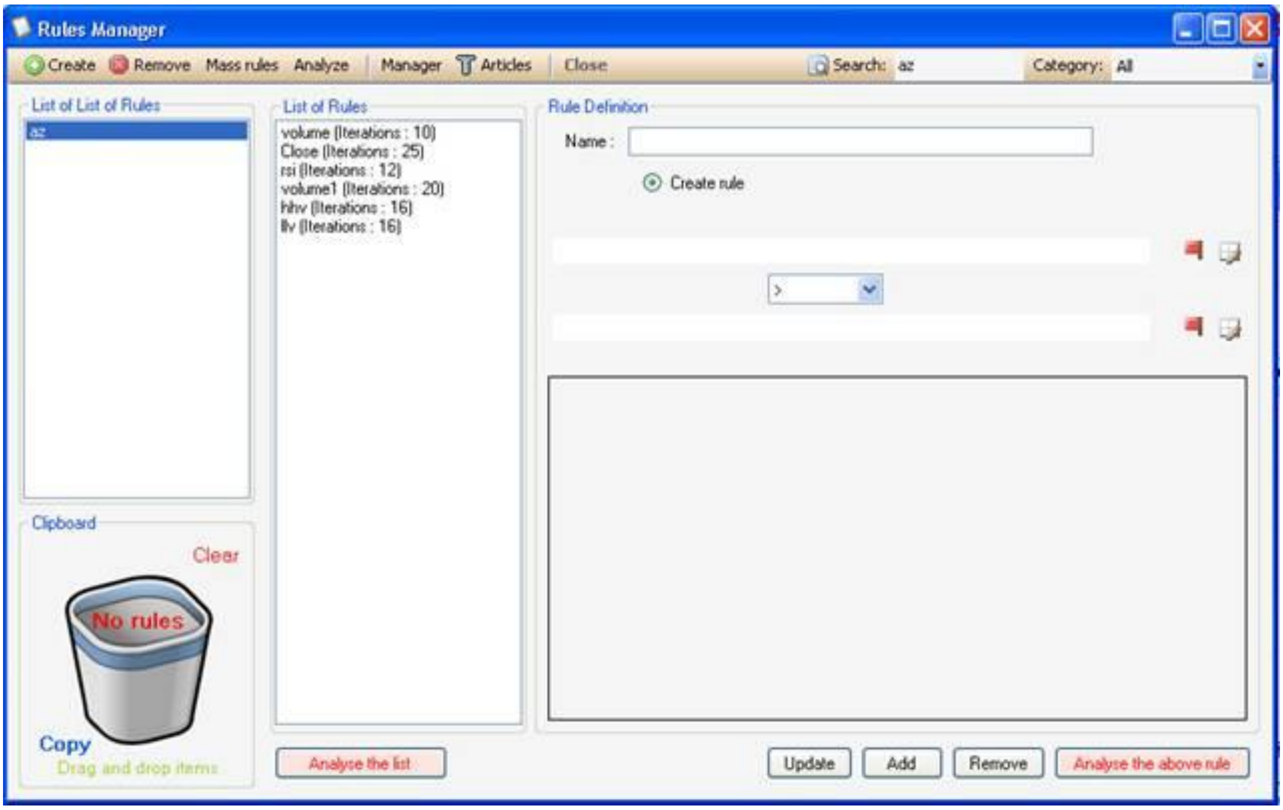
```
rsi(14)
>
a
```

Variable	Min	Max	Step
a	40	70	10

Type in the minimum, the maximum and the step value
In this example, we set a min of 50, a max of 90 and a step of 10, and this rule became a five iterations rule.

1.3. Rules Manager

Open the rule manager (*Analysis -> Rules Manager*)



Search: az

Category: All

List of List of Rules: az

List of Rules:

volume (Iterations : 10)
Close (Iterations : 25)
rsi (Iterations : 12)
volume1 (Iterations : 20)
hhv (Iterations : 16)
llv (Iterations : 16)

Name :

● Create rule

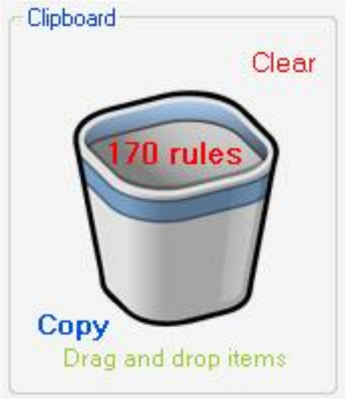
>

- Add: Create a list of rules, a dialog box will show up, you just have to enter a name and click on OK

- Remove: Select a list of rules from the list of rules List then click on "Remove" to delete this list
- Analyze: Analyze a list of rules (See below)
- Mass Rules: Create mass rules (See below)
- Close: Close the 'Rules Manager' form

1.4. Drag and drop

You can drag and drop rules or list of rules by selecting a rule or a list of rules and dragging it to the clipboard. From the clipboard, you can drag the posted rules by clicking on 'copy'.



1.5. Mass Rules

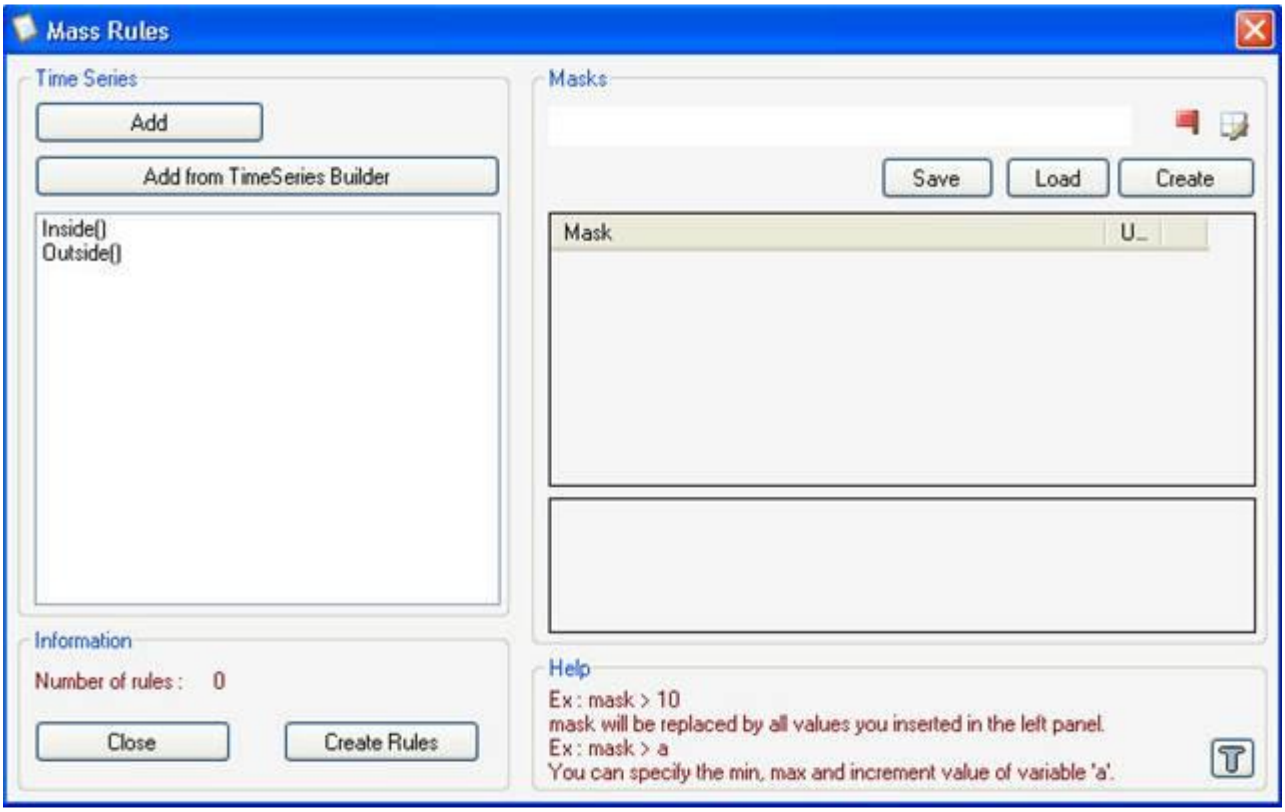
'Mass rules' is a tool that will help you create many rules very quickly using masks.

Click on 'Mass Rules' button; a dialog box will appears.

Now you have two ways to add functions:

- Add button
- Add from time-series button

The added functions will appear on the left list.
In our example, we added two functions: Inside() and Outside()



Inside()
Outside()

Number of rules : 0

Ex : mask > 10
mask will be replaced by all values you inserted in the left panel.
Ex : mask > a
You can specify the min, max and increment value of variable 'a'.

You can now create your masks and specify variables within masks if needed.

Type for example: mask > ref(mask, a) then click on Create, and make variable "a" vary from one to 10 with one as a step. We are about now to create 20 rules (2 rules and ten iterations for each rule)

After clicking on "Create Rules", two rules will be created (with 10 iterations for each rule)

```
Inside() > ref(Inside(), a)
Outside() > ref(Outside(), a)
```

1.6. Analyze rules

Analyze rules is a tool that will let you back-test your rules one by one before using them in ranking systems, simulation, neural networks...

The rules analyzer is not limited to analyzing the performance of trading rules, it can also be used to analyze any market behavior that occur after the trading rule signal become valid.
If you are trading options, you can for example analyze which rules or combination of rules lead to a security higher volatility.

Select a list of rules then click on Analyze, the settings dialog box will appear.

1.6.1. Symbols & Periods

Specify the symbols and the simulation period.

1.6.2. Filter

Specify upper and lower limits for a position output; every position with an output value outside these limits will be

rejected.
You can also specify limits by selecting a maximum number of standard deviation for position outputs.

1.6.3. Outputs

Outputs are formulas that the back-tester calculates when he enters a new position. Example: Performance for the next 10 bars which corresponds to the formula "perf(open, -10)"

The outputs form appears once you click on the "Select Outputs" button.

Defined outputs:
Performance, use a trailing stop: Enter a position then exit when drawdown reach –X Percent (Trailing stop)
Performance, use N-Bars stop: Enter a position then exit after X bars (N-Bars stop)
Performance, custom rule for exit + trailing stop: Enter a position then exit when drawdown reach –X Percent or when the specified rule is TRUE.
Performance, exit when rule is false: Enter a position then exit when the rule is no longer TRUE or valid.

The "Custom Output" lets you add your own advanced formula.
See (HELP: Advanced Rules; Output) for an example

1.6.4. Fixed Rule

You can specify a fixed rule that will be added to the rules to be analyzed.
Fixed rule can contain iterations.
Example:

Let us say we want to analyze theses rules:

- RSI (14) > 10
- Perf (close, 10) > a, where "a" varies from two to four, with a step of two

As a fixed rule, we have:
"close > cl" , where "cl" varies from 10 to 20 with a step of 10.

You will end up with a total of "1 + (1 * 2)" rules multiplied by 2 (fixed rules) for a total number of six rules.
Here is the list of all the rules that will be analyzed:

close > 10 && Rsi(14) > 10

close > 10 && Perf(close, 10) > 2

close > 10 && Perf(close, 10) > 4

close > 20 && Rsi(14) > 10

close > 20 && Perf(close, 10) > 2

close > 20 && Perf(close, 10) > 4

1.6.5. Metrics

CSharp or JScript.Net languages are used to create metrics.

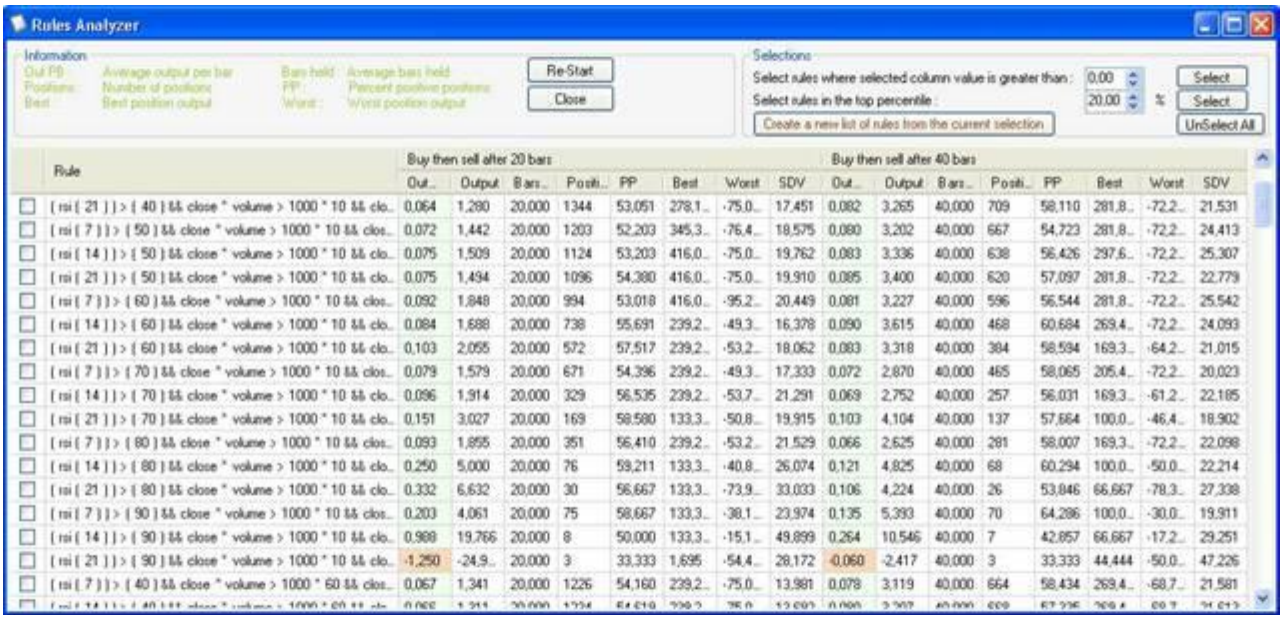
Example of metric formula:

Fitness = OutputPerBar * (NbPositions > 100);

Set the output to zero if the number of position generated by the rules is lower than one hundred.
For each "metric formula", you can add additional metrics using the following function: Functions.AddMetric

1.6.6. Results

When analyzing rules completes, a grid containing the results appears.



Rules Analyzer
Information
Out PB : Average output per bar
Positions : Number of positions
Best : Best position output
Bars held : Average bars held
PP : Percent positive positions
Worst : Worst position output
Re-Start
Close
Selections
Select rules where selected column value is greater than : 0,00
Select rules in the top percentile : 20,00 %
Select
Select
UnSelect All
Create a new list of rules from the current selection

	Rule	Buy then sell after 20 bars								Buy then sell after 40 bars							
		Out...	Output	Bars...	Positi...	PP	Best	Worst	SDV	Out...	Output	Bars...	Positi...	PP	Best	Worst	SDV
☐	(rsi (21)) > (40) && close * volume > 1000 * 10 && clo...	0,064	1,280	20,000	1344	53,051	278,1...	-75,0...	17,451	0,082	3,265	40,000	709	58,110	281,8...	-72,2...	21,531
☐	(rsi (7)) > (50) && close * volume > 1000 * 10 && clos...	0,072	1,442	20,000	1203	52,203	345,3...	-76,4...	18,575	0,080	3,202	40,000	667	54,723	281,8...	-72,2...	24,413
☐	(rsi (14)) > (50) && close * volume > 1000 * 10 && clo...	0,075	1,509	20,000	1124	53,203	416,0...	-75,0...	19,762	0,083	3,336	40,000	638	56,426	297,6...	-72,2...	25,307
☐	(rsi (21)) > (50) && close * volume > 1000 * 10 && clo...	0,075	1,494	20,000	1096	54,380	416,0...	-75,0...	19,910	0,085	3,400	40,000	620	57,097	281,8...	-72,2...	22,779
☐	(rsi (7)) > (60) && close * volume > 1000 * 10 && clos...	0,092	1,848	20,000	994	53,018	416,0...	-95,2...	20,449	0,081	3,227	40,000	596	56,544	281,8...	-72,2...	25,542
☐	(rsi (14)) > (60) && close * volume > 1000 * 10 && clo...	0,084	1,688	20,000	738	55,691	239,2...	-49,3...	16,378	0,090	3,615	40,000	468	60,684	269,4...	-72,2...	24,093
☐	(rsi (21)) > (60) && close * volume > 1000 * 10 && clo...	0,103	2,055	20,000	572	57,517	239,2...	-53,2...	18,062	0,083	3,318	40,000	384	58,594	169,3...	-64,2...	21,015
☐	(rsi (7)) > (70) && close * volume > 1000 * 10 && clos...	0,079	1,579	20,000	671	54,396	239,2...	-49,3...	17,333	0,072	2,870	40,000	465	58,065	205,4...	-72,2...	20,023
☐	(rsi (14)) > (70) && close * volume > 1000 * 10 && clo...	0,096	1,914	20,000	329	56,535	239,2...	-53,7...	21,291	0,069	2,752	40,000	257	56,031	169,3...	-61,2...	22,185
☐	(rsi (21)) > (70) && close * volume > 1000 * 10 && clo...	0,151	3,027	20,000	169	58,580	133,3...	-50,8...	19,915	0,103	4,104	40,000	137	57,664	100,0...	-46,4...	18,902
☐	(rsi (7)) > (80) && close * volume > 1000 * 10 && clos...	0,093	1,855	20,000	351	56,410	239,2...	-53,2...	21,529	0,066	2,625	40,000	281	58,007	169,3...	-72,2...	22,098
☐	(rsi (14)) > (80) && close * volume > 1000 * 10 && clo...	0,250	5,000	20,000	76	59,211	133,3...	-40,8...	26,074	0,121	4,825	40,000	68	60,294	100,0...	-50,0...	22,214
☐	(rsi (21)) > (80) && close * volume > 1000 * 10 && clo...	0,332	6,632	20,000	30	56,667	133,3...	-73,9...	33,033	0,106	4,224	40,000	26	53,846	66,667	-78,3...	27,338
☐	(rsi (7)) > (90) && close * volume > 1000 * 10 && clos...	0,203	4,061	20,000	75	58,667	133,3...	-38,1...	23,974	0,135	5,393	40,000	70	64,286	100,0...	-30,0...	19,911
☐	(rsi (14)) > (90) && close * volume > 1000 * 10 && clo...	0,988	19,766	20,000	8	50,000	133,3...	-15,1...	49,899	0,264	10,546	40,000	7	42,857	66,667	-17,2...	29,251
☐	(rsi (21)) > (90) && close * volume > 1000 * 10 && clo...	-1,250	-24,9...	20,000	3	33,333	1,695	-54,4...	28,172	-0,060	-2,417	40,000	3	33,333	44,444	-50,0...	47,226
☐	(rsi (7)) > (40) && close * volume > 1000 * 60 && clos...	0,067	1,341	20,000	1226	54,160	239,2...	-75,0...	13,981	0,078	3,119	40,000	664	58,434	269,4...	-68,7...	21,581

Each line represents a rule or iteration.

Columns abbreviations:

- Out PB: Average position outputs per bar
- Output: Average position outputs
- Bars held: Average number of bars per position
- Positions: Number of positions
- PP: Percentage of positive position output
- Best: Best position output
- Work: Worst position output
- SDV: Standard deviation of position outputs

You can easily create new list of rules from the analyzed rules.

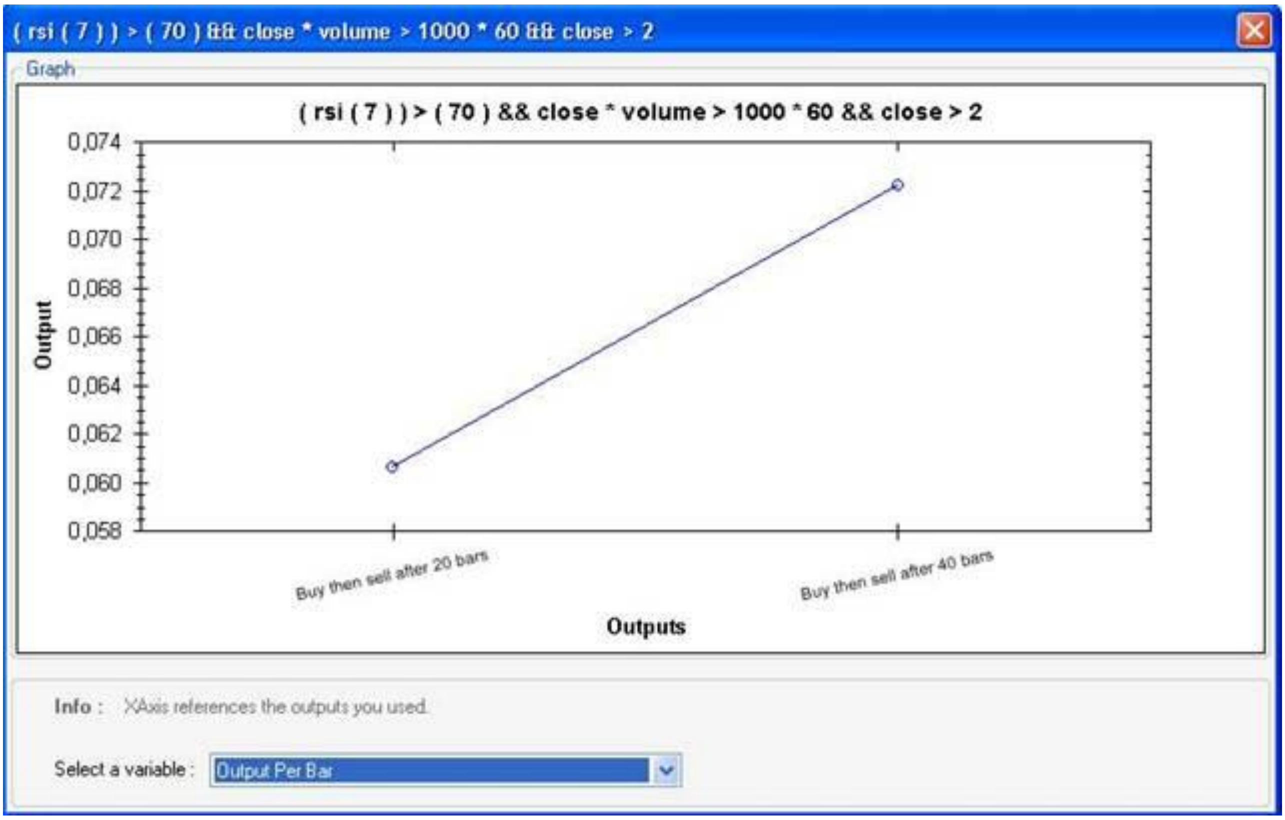
On the "Rules" column click on the right mouse button to make a menu appears.

Search:

Search for rules that contain a specific keyword, found rules will be checked.

Graph per Output:

Displays a chart that plots the analyzer results per output. The average of all rule outputs for each output type will be calculated.

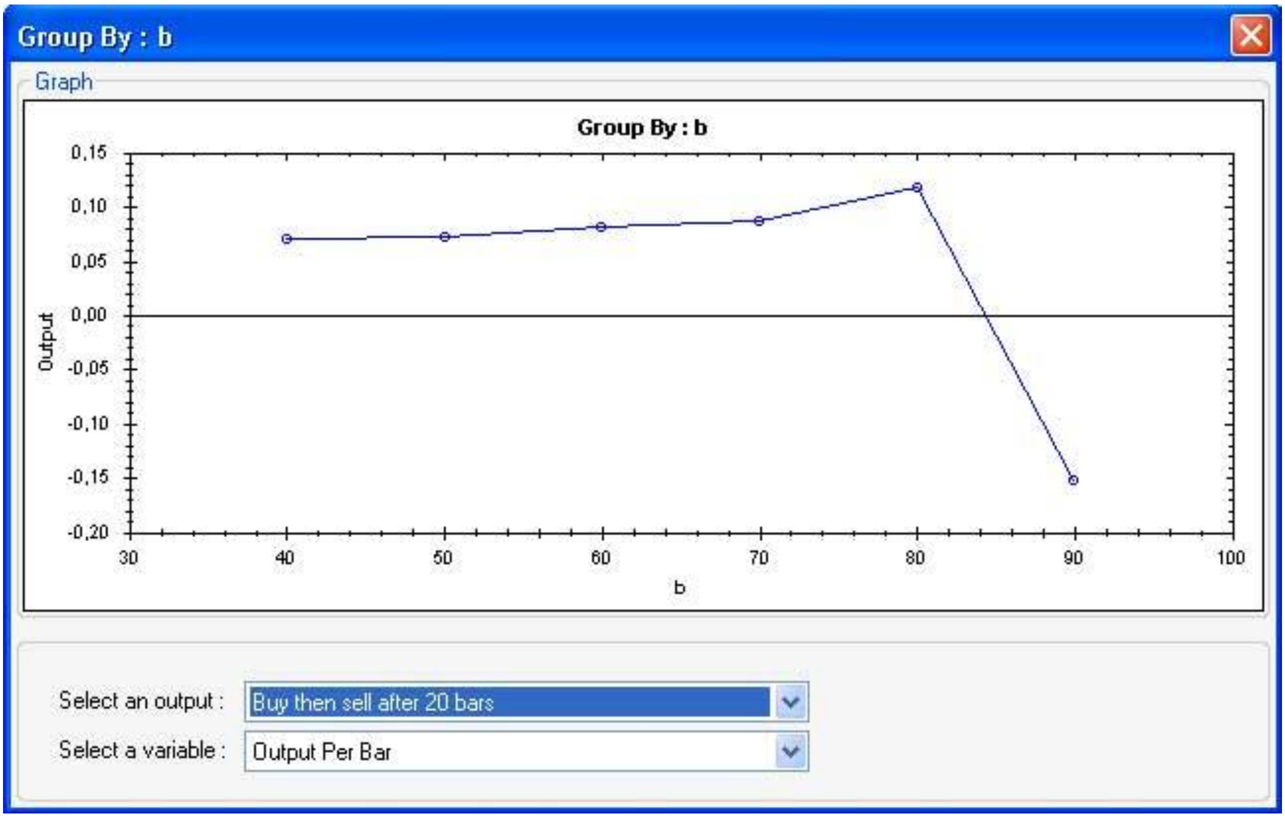

(rsi (7)) > (70) && close * volume > 1000 * 60 && close > 2
Graph
(rsi (7)) > (70) && close * volume > 1000 * 60 && close > 2
0,074
0,072
0,070
0,068
0,066
0,064
0,062
0,060
0,058
Output
Buy then sell after 20 bars
Buy then sell after 40 bars
Outputs
Info :

XAxis references the outputs you used.

Select a variable : Output Per Bar

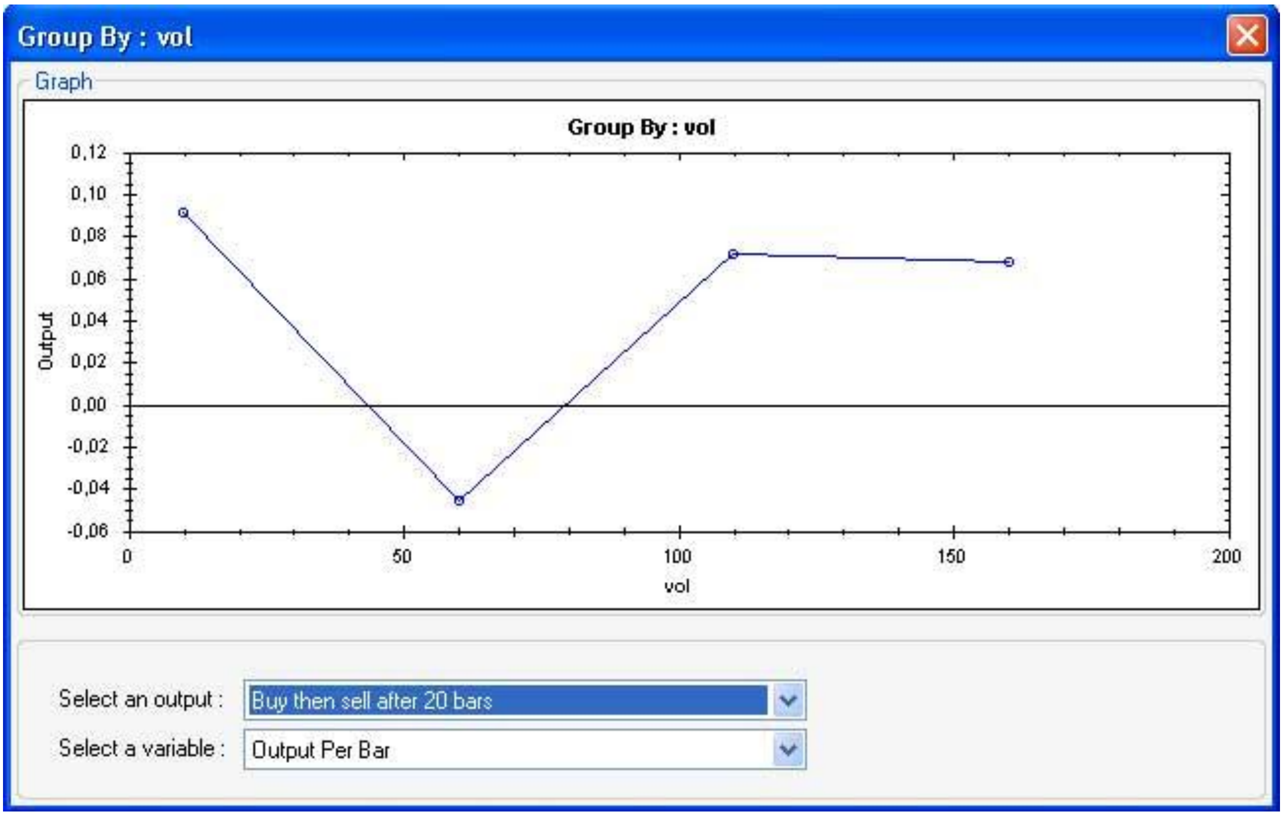
Group results per variable:

Displays a chart that plots the analyzer results per variable value. The average of all rule outputs for each variable value will be calculated.



Select an output : Buy then sell after 20 bars

Select a variable : Output Per Bar



Select an output : Buy then sell after 20 bars

Select a variable : Output Per Bar

[TOP]

1.2.6.10.2 Ranking System Manager

1. Ranking System Manager

1.1. Introduction

A Ranking System takes a list of symbols and assigns a value to each symbol within this list.
Generally, a symbol value ranges from zero to 100.

1.2. Create a ranking system

To create a ranking system, open the ranking system manager, *Analysis->Ranking System Manager*, click on the 'Add' button and then type a name.
Now select the ranking system you have just created and start adding formulas and nodes.

Add parent node: Add a top node
Add node: Add a node within the selected node
Add formula: Add a formula within the selected node

Nodes and formula have in common: the name and the weight properties.
Right click on a node or formula to open a menu.

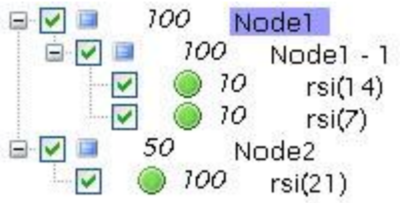
You can assign a different weight to each node or formula to give them more or less importance.

Formulas have some others properties:

- **Formula:** a rule or formula you want to use. The red flag nears the formula input box and the circle in the tree view indicates whether the formula is correct (green) or incorrect (red).
- **Factor:** multiply the formula result by the provided value
- **Description:** the description of the formula
- **Maximum and minimum levels:** The maximum and minimum value the formula can have, if a formula return 110 while the maximum level is set to 100, then the application will use a value of 100.

Auto update factor: If this item is checked, the application will decide which factor to apply to the formula. The chosen value is not always accurate, so it is better to update the 'factor' field manually. Generally if the formula return values between 0 and 100, for example an indicator like the relative strength index (rsi), then a factor of one should be used; if the formula return either zero or one (Ex : close > 20), then a factor of 100 should be used.

Calculation example:



```
⊟ ☑ 100 Node1
   ⊟ ☑ 100 Node1 - 1
      ☑ ● 10 rsi(14)
      ☑ ● 10 rsi(7)
⊟ ☑ 50 Node2
   ☑ ● 100 rsi(21)
```

Let us make some calculation using this above ranking system as an example.
As you can see, this ranking system contains two nodes.
The first node contains one node which contains two formulas.
The second node contains one formula.

The calculation begins with the formulas.

Formula: rsi (14)

Bars	1	2	3	4	5
Values	45	46	50	52	54

Formula: rsi (7)

Bars	1	2	3	4	5
Values	20	23	29	20	19

Formula: rsi (21)

Bars	1	2	3	4	5
Values	60	61	60	50	45

The values equals the rsi value multiplied by the factor value of the formula, we took here a factor value of one.
Now the software will calculate the parent node of theses formulas.
For each bar, the software takes the value of each node's children (formula or node), multiply it by the formula or node's weight, add all the values, and then divide the result by the sum of the children's weights.

For the bar number one and for the node 'Node1 – 1'
S = sum of children's weights: 10 + 10 = 20

rsi(14) value = 45 * 10 = 450
rsi(7) value = 20 * 10 = 200
rsi(14) value + rsi(7) value = 450 + 200 = 650
'Node 1 – 1' value = 650 / 20 = 32.5.

Node: Node1 - 1

Bars	1	2	3	4	5
Values	32.5	34.5	39.5	36	36.5

We make the same calculation for the 'Node2'.
Because the 'Node2' has only one child, it will get the same values as its child.

Node: Node2

Bars	1	2	3	4	5
Values	60	61	60	50	45

Node: Node1 (same as Node1 – 1)

Bars	1	2	3	4	5
Values	32.5	34.5	39.5	36	36.5

Now, the software will calculate the 'Top Node' (Ranking system values) which has two children (Node1 and Node2).
The same calculation as described above applies.
Top Node = ((Node1 * Node1_weight) + (Node2 * Node2_weight)) / (Node1_weight + Node2_weight).

Therefore, for the first bar it will be:

Top Node bar1 = ((32.5 * 100) + (60 * 50)) / (100 + 50)

Top Node: (Ranking system values)

Bars	1	2	3	4	5
Values	41.67	43.33	46.33	40.67	39.33

NB:
If you want to ignore a node or formula you can uncheck it, only checked nodes are evaluated.

1.3. Create a ranking system from a list of rules

Select a node, right click on it and then click on 'Add rules from list of rules'.
You just have to select a list of rules then your ranking system will be populated with these rules.

1.4. Analyze Ranking System

To analyze or back-test a ranking system, click on 'Analyze Ranking System'.
The 'Analyze Ranking System Settings' form will appears, update settings then click on 'Analyze'.

The ranking system analyzer settings are the same as in the 'Analyze List of Rules Settings'.
However, there are two more fields:

- **Number of buckets:** Let you specify the number of buckets you want to create
  Let us say, our ranking system returns values from zero to 100, if we specify 10 as the number of buckets, then the application will back-test or analyze ten systems.
  The first system will be the one that have values between 0 and 10, the next one with values between ten and 20... (If you choose Normal Ranking as a Ranking Calculation)

- **Ranking calculation**:

  Same number of positions in each bucket: The boundary between states are choosen so that each state is assigned the same number of positions.

  Normal Ranking: The boundary between states are choosen so all the states have the same size.

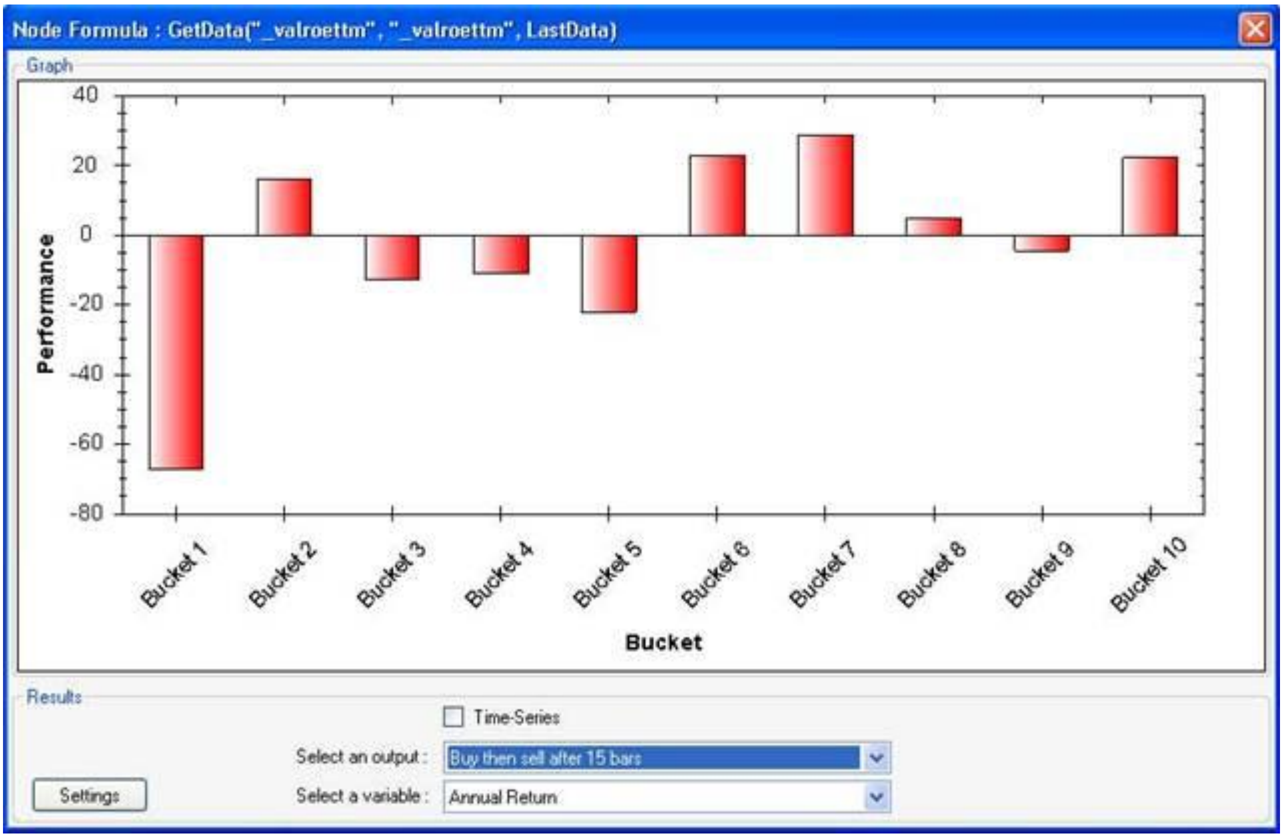
Example: number of buckets = 10; bucket 1 will contains all the positions that have a rank between zero and ten, bucket 2 will contains all the positions that have a rank between ten and twenty and so on.

Ranking on each rebalance date: For each rebalance period, calculate the values, sort symbols by their value and assign a rank to each symbol (distance between the ranks will be equal).
Example: For a specific date, we have four symbols with the following ranks: 11, 10, 20, and 90; we first sort these symbols then assign a new rank to each one, we will end up with the following new ranks: 0, 25, 50, and 100, where symbol number one have a rank of 25, symbol number two have a rank of 0 and so on.
(The trading system back-tester uses this type of ranking)

- **Rebalance every:** The analyzer starts at the start date and advance by a number of days as specified in this field. For each period, a rebalance is done (old positions are closed and new ones are taken).
Unlike the 'Rules Analyzer', which rebalance or taking a new position occur only after the old position is closed, the 'Ranking System Analyzer' takes new positions after a specific number of days; this position is then assigned to a bucket depending on the value of the ranking system at this date.

1.5. Analyze Ranking System Report

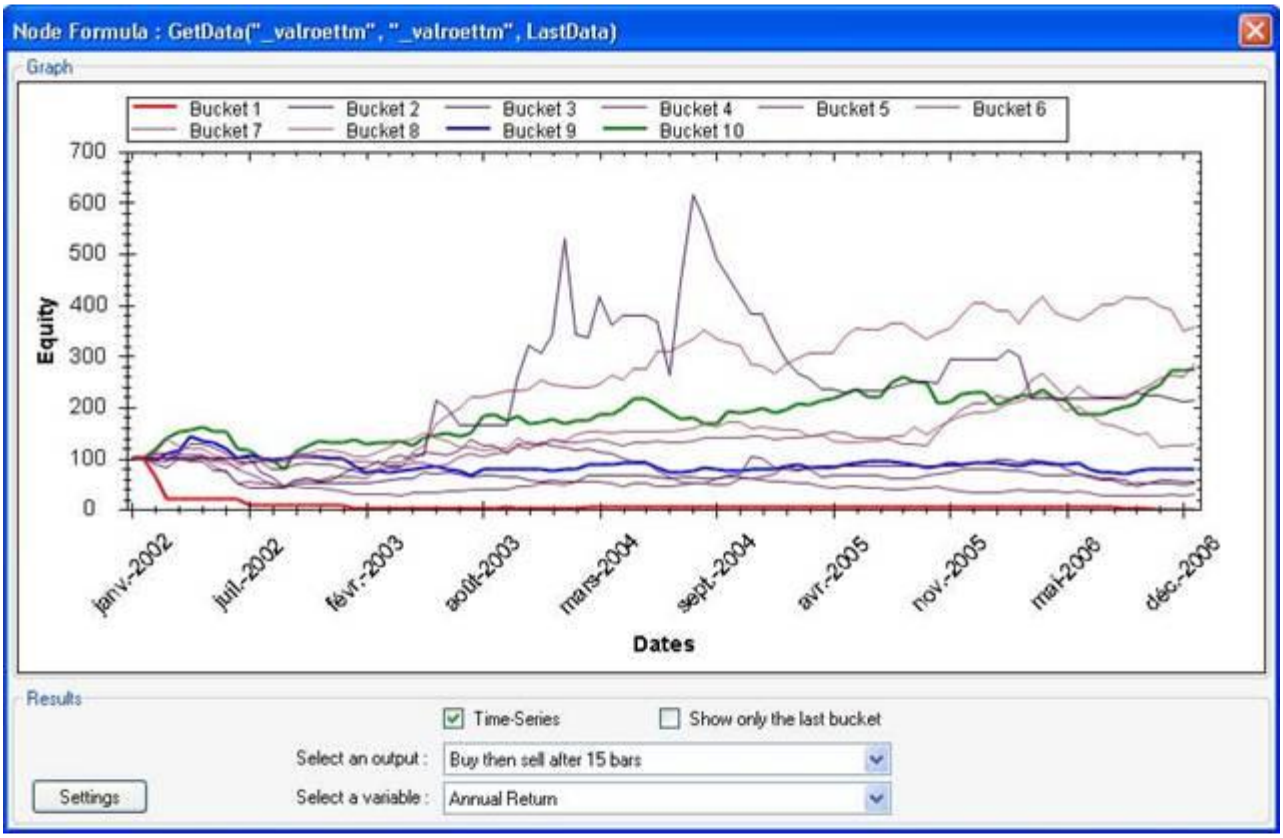
The 'Analyze Ranking System Report' contains all the back-testing results related to your ranking system.


Node Formula : GetData("_valroettm", "_valroettm", LastData)

□ Time-Series

Select an output : Buy then sell after 15 bars

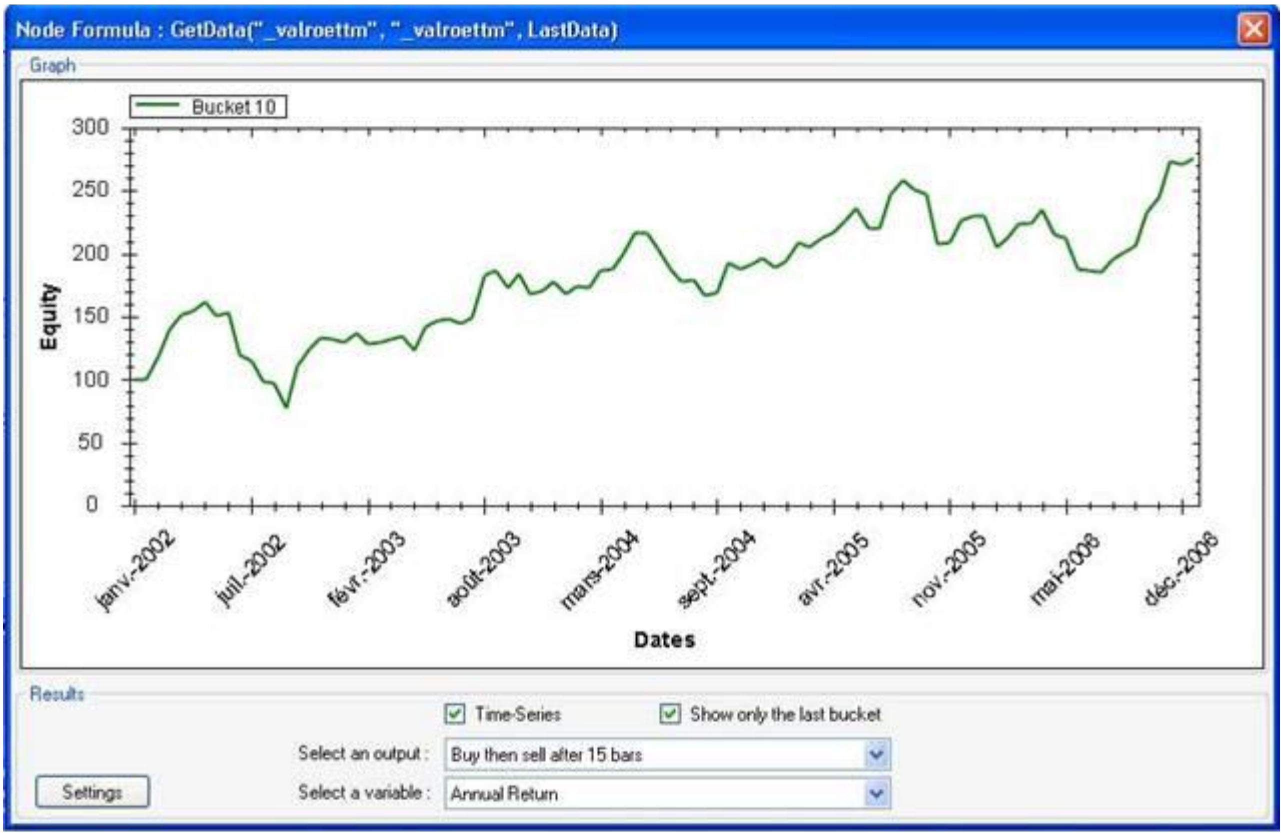
Select a variable : Annual Return



☑ Time-Series ☐ Show only the last bucket

Select an output : Buy then sell after 15 bars

Select a variable : Annual Return



☑ Time-Series ☑ Show only the last bucket

Select an output : Buy then sell after 15 bars

Select a variable : Annual Return

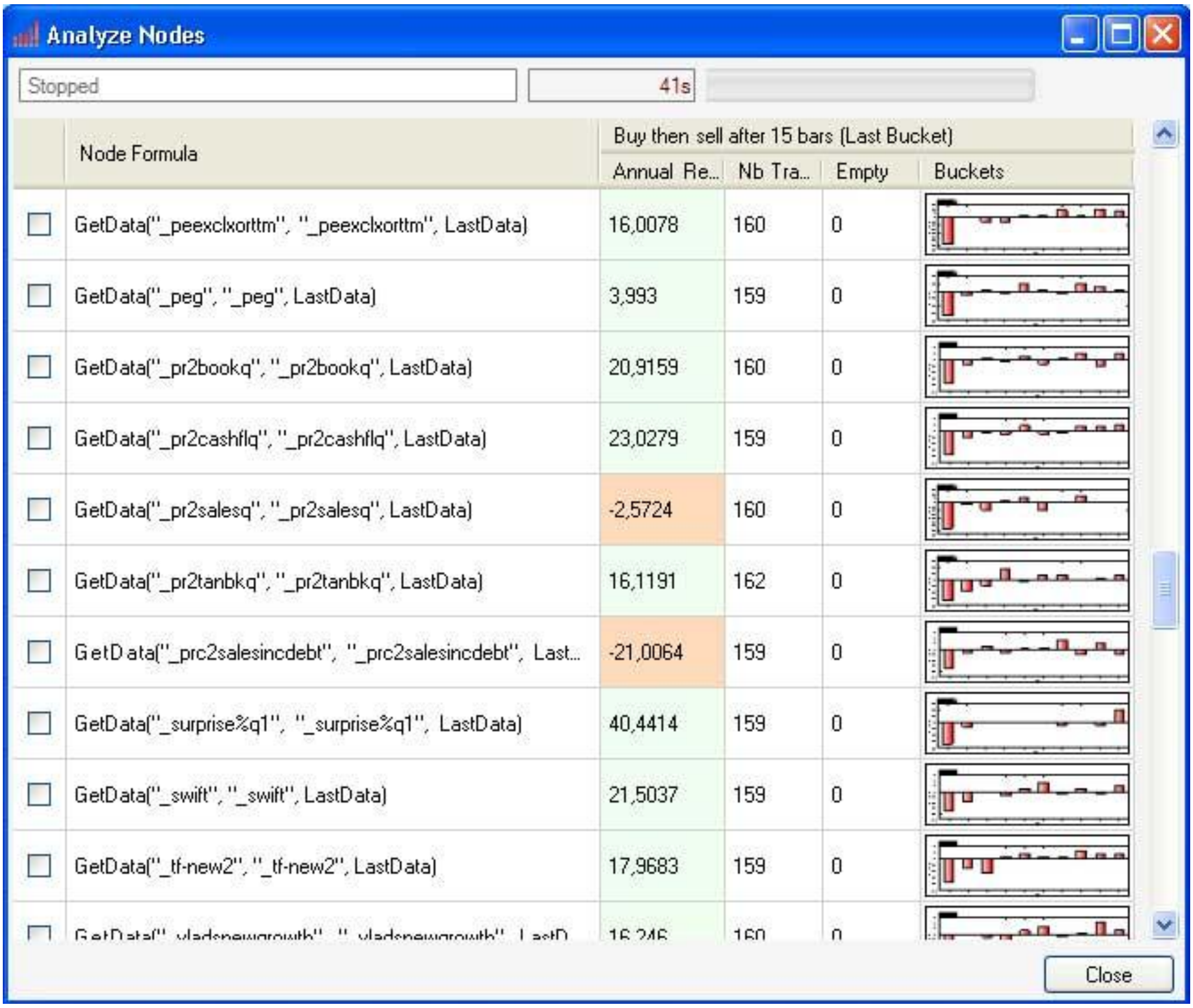
1.6. Analyze Nodes

Instead of analyzing a ranking system, you can also analyze each node of a ranking system separately; this will let you know which nodes perform well.
To analyze ranking system nodes, select a ranking system then click on 'Analyze Nodes'.

1.7. Analyze Nodes Report

The 'Analyze Nodes Report' contains all the back-testing results related to your ranking system nodes.

Each line represents a node, as you can see in the picture below.



	Node Formula	Buy then sell after 15 bars (Last Bucket)			
		Annual Re...	Nb Tra...	Empty	Buckets
☐	GetData("_peexclxorttm", "_peexclxorttm", LastData)	16,0078	160	0	
☐	GetData("_peg", "_peg", LastData)	3,993	159	0	
☐	GetData("_pr2bookq", "_pr2bookq", LastData)	20,9159	160	0	
☐	GetData("_pr2cashflq", "_pr2cashflq", LastData)	23,0279	159	0	
☐	GetData("_pr2salesq", "_pr2salesq", LastData)	-2,5724	160	0	
☐	GetData("_pr2tanbkq", "_pr2tanbkq", LastData)	16,1191	162	0	
☐	GetData("_prc2salesincdebt", "_prc2salesincdebt", Last...	-21,0064	159	0	
☐	GetData("_surprise%q1", "_surprise%q1", LastData)	40,4414	159	0	
☐	GetData("_swift", "_swift", LastData)	21,5037	159	0	
☐	GetData("_tf-new2", "_tf-new2", LastData)	17,9683	159	0	
☐	GetData("_vladsnewgrowth", "_vladsnewgrowth", LastD	16,246	160	0	

Click on the charts to display additional back-testing information.

1.8. Ranking System & Formulas

You can access a ranking system from the QuantShare language by using this function:

```
Ranking("ranking system category", "ranking system name");
```

This function will return the ranking system value of the current symbol. You can use it to create rules and then apply them to the rules analyzer, ranking system analyzer (Ranking system based on another ranking system), trading systems and prediction models...

[TOP]

1.2.6.10.3 Simulator

1. Simulator

1.1. Introduction

The Simulator lets you test strategies or trading systems under historical market conditions to determine whether certain scenarios would have worked well in the past. The idea is that if a trading strategy would have performed well previously, it may be worth considering today.

A trading system includes:

- Buy rules
- Sell rules
- Short rules
- Cover rules
- Capital settings
- Stops
- Money management rules
- Long and short ranking systems

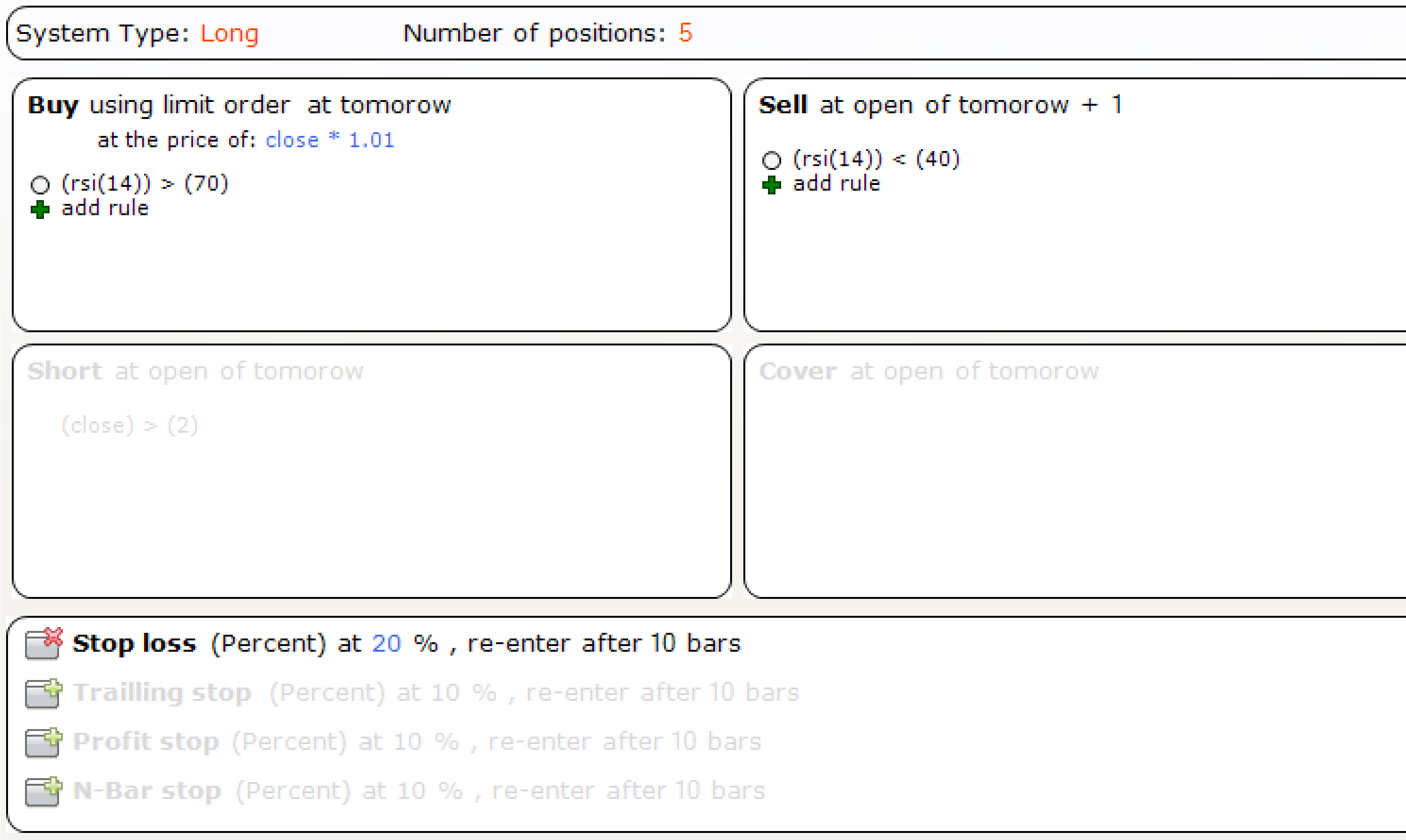
1.2. Create a Trading System

Open the 'Simulator' (*Analysis->Simulator*)

- Click on 'New'
- Set entries and exits
- Set the capital settings
- Set stops
- Set money management rules
- Set symbols that are going to be used in this trading system
- Set a ranking system

Finally, click on 'Create trading system'.

1.4. Entries, Exits and Stops



System Type: Long Number of positions: 5

Buy using limit order at tomorow
at the price of: close * 1.01

○ (rsi(14)) > (70)
✚ add rule

Sell at open of tomorow + 1

○ (rsi(14)) < (40)
✚ add rule

Short at open of tomorow

(close) > (2)

Cover at open of tomorow

Stop loss (Percent) at 20 % , re-enter after 10 bars

Trailling stop (Percent) at 10 % , re-enter after 10 bars

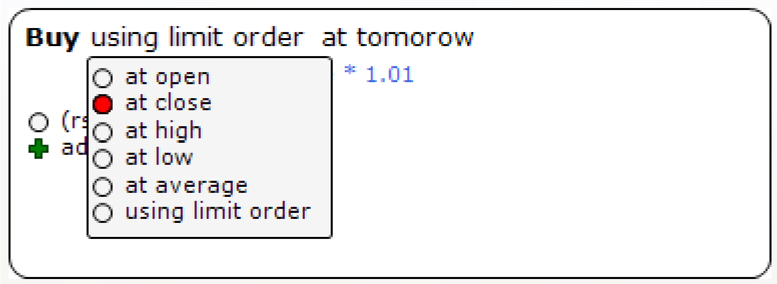
Profit stop (Percent) at 10 % , re-enter after 10 bars

N-Bar stop (Percent) at 10 % , re-enter after 10 bars

System Type: specify whether to create a long, a short or a long/short trading system

Number of positions: Specify the maximum number of positions in the portfolio. You can use the advanced money management script to dynamically change the number of positions in your portfolio.

Entries and exits settings: If a label is underlined when the mouse is over it, click on the mouse left button to display a small menu. You can also update values directly using the mouse wheel.



Buy using limit order at tomorow

* 1.01

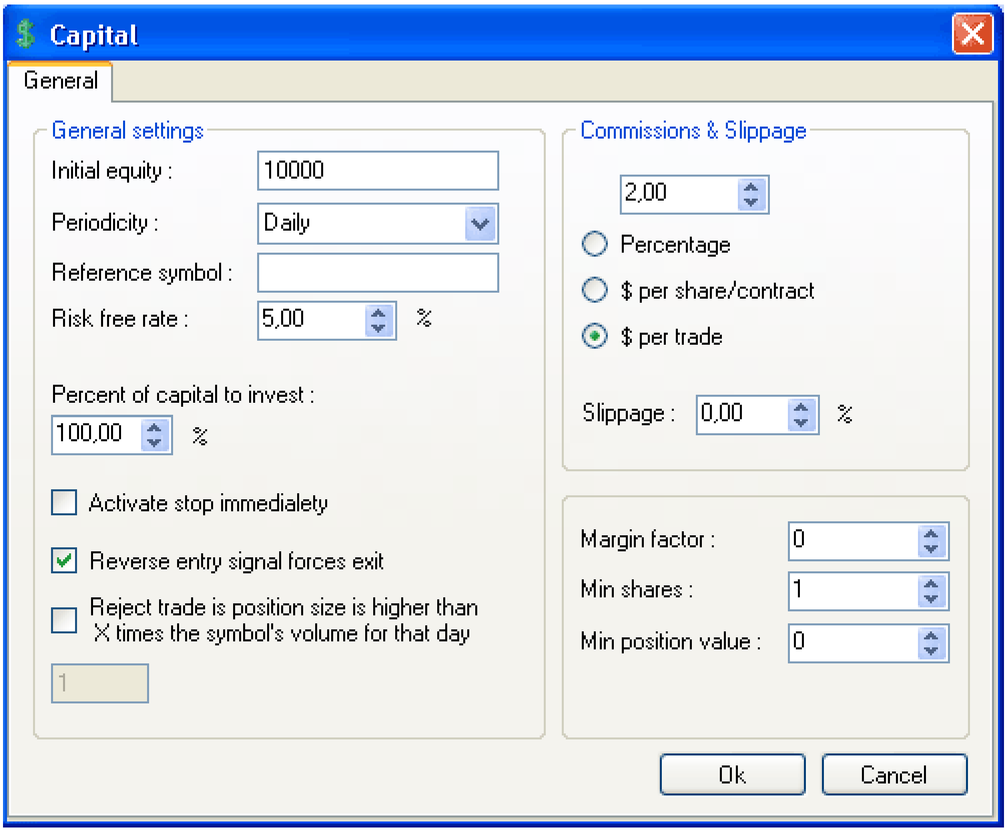
○ at open
● at close
○ at high
○ at low
○ at average
○ using limit order

Entries and exits rules: Click on a rule or on "add rule" to update, add, remove and manage rules.

Specify the stops to apply to the trading system by clicking on the appropriate icon:

- Maximum loss stop: Exit a position if the symbol's performance (points or percent) falls below this level
- Profit target stop: Exit a position if the symbol's performance reaches this level
- Trailing stop: Exit a position if the symbol's drawdown falls below this level
- N-bar stop: Exit a position after a specific number of bars

1.3. Capital Settings



General settings

Initial equity : 10000

Periodicity : Daily

Reference symbol :

Risk free rate : 5,00 %

Percent of capital to invest : 100,00 %

□ Activate stop immedialety

☑ Reverse entry signal forces exit

□ Reject trade is position size is higher than X times the symbol's volume for that day

1

Commissions & Slippage

2,00

○ Percentage

○ $ per share/contract

● $ per trade

Slippage : 0,00 %

Margin factor : 0

Min shares : 1

Min position value : 0

The Capital settings include:

- Initial equity
- Periodicity: rebalance period
- Reference symbol: generally an index like the SP500
- Risk free rate
- Percent of capital to invest: Specify the percentage of capital to invest
- Activate stop immediately: If activated then when exiting a trade because of a stop, uses the stop value as the trade exit price, otherwise exit as defined in the sell or cover price and delay settings)
- Reverse entry forces exit: When checked, the back-tester will close a long position if he encounter a short signal for the same symbol and close a short position if he encounters a long signal for the same symbol.
- Reject trade if position size is higher than x times the symbol's volume for that day: tell the back-tester to enter position only if there is enough liquidity.
- Margin: a factor to specify the margin you want to allow for this trading system (A value of 2 means that the amount of money you are allowed to use within this trading system is twice your current equity value)
- Minimum shares: This is the minimum number of shares you can buy or short. (The position will not take place if the number of shares to buy or short is less that this value)
- Minimum position value: The same as 'Minimum shares', but we use here the position value or volume.
- Commissions: Brokerage commissions
- Slippage: specify the average slippage associated with each trade. (Difference between the simulated entry price and the real entry price)
- Allow penny stocks: Reject a trade or a position if the stock price is lower than 2. Use this field only if you are trading stocks.

1.5. Money Management

See Advanced Money Management

1.6. Optimize a trading system

You can optimize a trading system using the optimizer plug-in or using the simulator optimizer.
The optimizer plug-in uses complex algorithms to find the best trading system, while the integrated simulator optimizer use bulk optimization (that is, the simulator will back-test all the possibilities)

1.7. Run or optimize a simulation

To run a simulation, first select a trading system, and then click on "Simulation".
To optimize a simulation, first select a trading system, and then click on "Optimize".

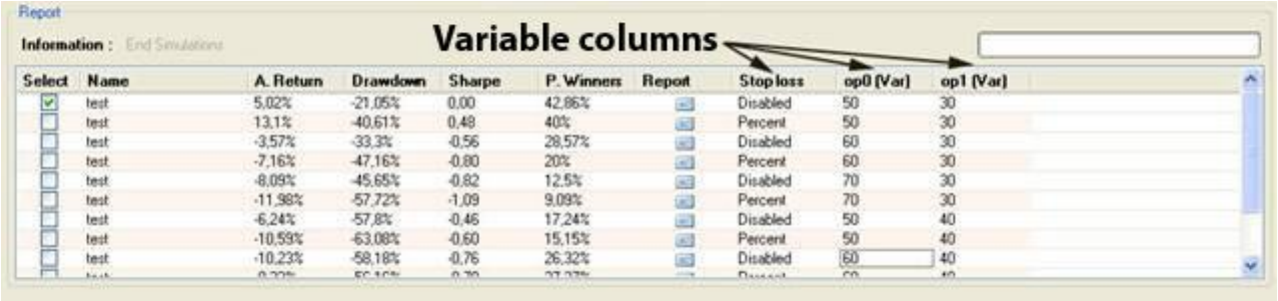
1.8. Trading System Report

1.8.1 Introduction

The "Trading System Report" analyzes all the signals generated by the simulation and gives you a detailed report.

1.8.2 Grid

The grid control shows you the simulation basic statistics.
When using optimization, this grid shows all generated simulations, a column for each variable shows you what value were used for each simulation.



Information : End Simulations

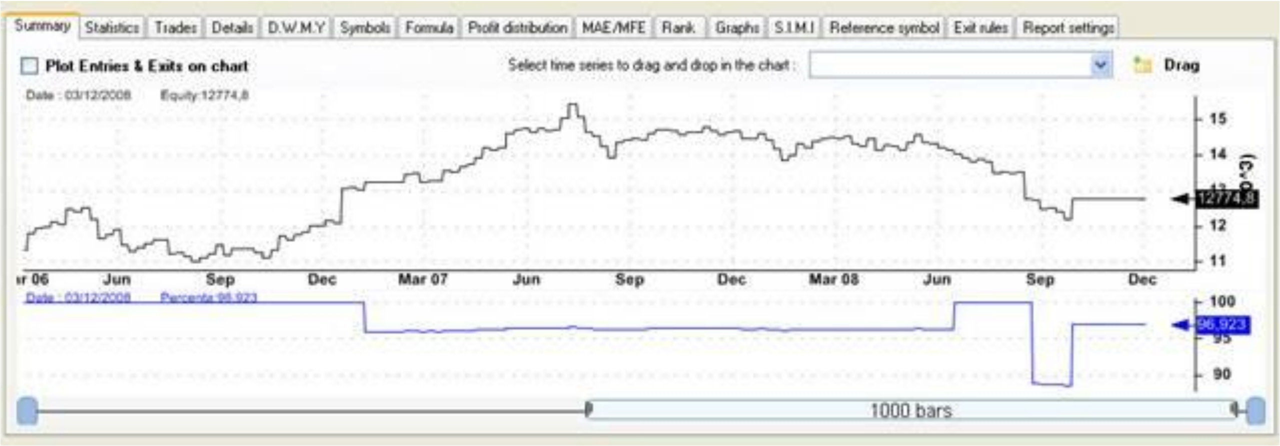
Select	Name	A. Return	Drawdown	Sharpe	P. Winners	Report	Stop loss	op0 (Var)	op1 (Var)
☑	test	5,02%	-21,05%	0,00	42,86%	▤	Disabled	50	30
□	test	13,1%	-40,61%	0,48	40%	▤	Percent	50	30
□	test	-3,57%	-33,3%	-0,56	28,57%	▤	Disabled	60	30
□	test	-7,16%	-47,16%	-0,80	20%	▤	Percent	60	30
□	test	-8,09%	-45,65%	-0,82	12,5%	▤	Disabled	70	30
□	test	-11,98%	-57,72%	-1,09	9,09%	▤	Percent	70	30
□	test	-6,24%	-57,8%	-0,46	17,24%	▤	Disabled	50	40
□	test	-10,59%	-63,08%	-0,60	15,15%	▤	Percent	50	40
□	test	-10,23%	-58,18%	-0,76	26,32%	▤	Disabled	60	40

Double click on the 'Report' column icon to open the report of the selected simulation.

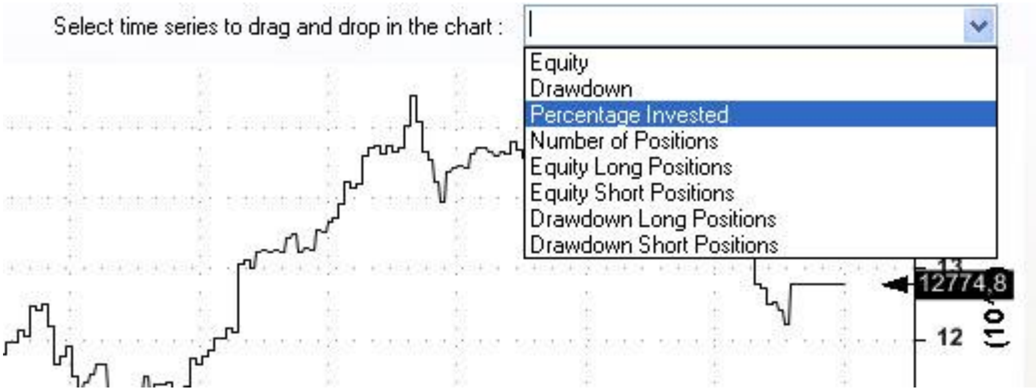
1.8.3 Chart

Plot equity, drawdown and others time-series graphs.

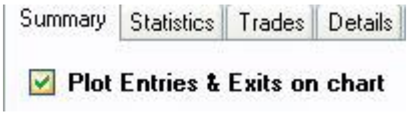
Right click on the chart to open a menu.
You can create new panes, remove panes, and remove a graph from a pane...



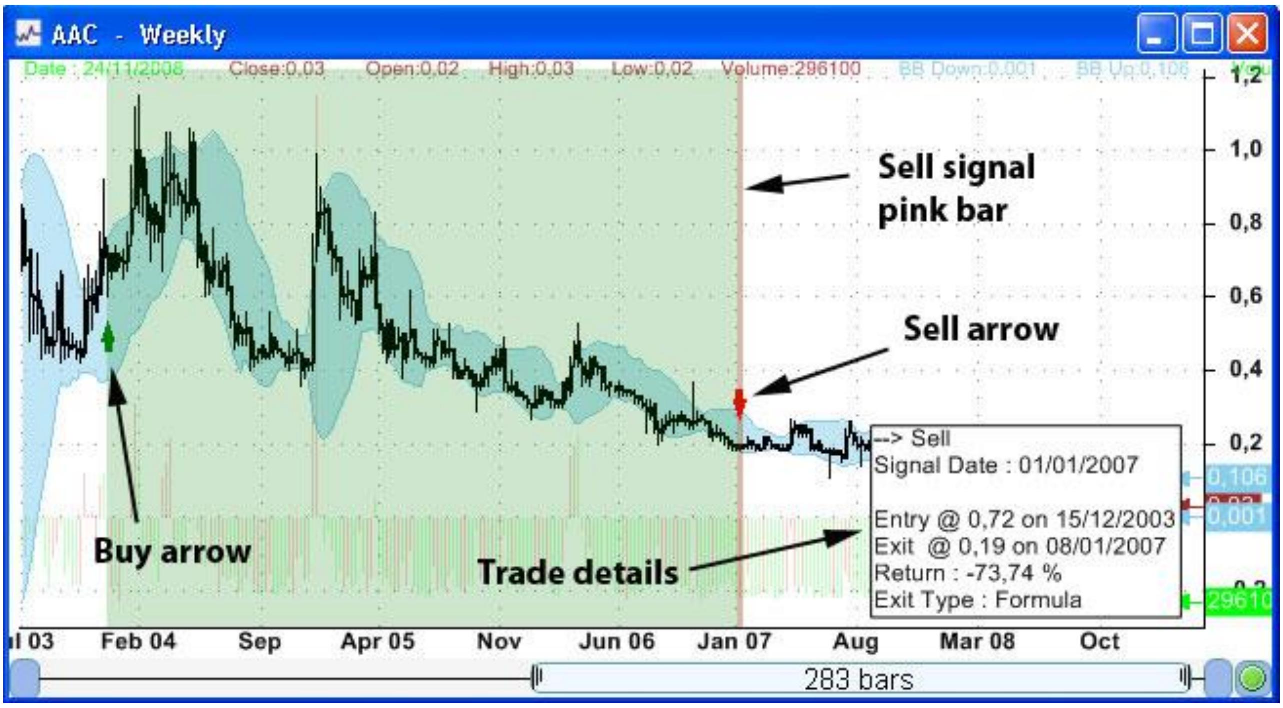
To add a graph to a pane, select a time-series from the combo box as shown below then click on 'Drag' icon then drag the time-series and drop it into a chart's pane.



To plot entry and exit signals in a symbol chart, check 'Plot Entries & Exits on chart' then select a symbol that was used in the simulation.



The symbol chart will look like the picture below. Click on the arrows to display the position details.



--> Sell
Signal Date : 01/01/2007

Entry @ 0,72 on 15/12/2003
Exit @ 0,19 on 08/01/2007
Return : -73,74 %
Exit Type : Formula

1.8.4 Statistics

Capital:

Initial Capital – The size of your account before the simulation started

End Capital – The size of your account after the simulation ended

Net Profit – The profit generated by the trading system

Net Profit in Percentage – The profit generated by the trading system in percentage

Exposure – The percentage of your account that was invested for all bars

Net Risk Adjusted Return – Net profit % divided by Exposure %

Annual Return – Compounded annual return %

Risk Adjusted Return - Annual return % divided by Exposure %

Average Profit/Loss - (Profit of winners + Loss of losers)/(number of trades)

Average Profit/Loss in Percentage - (% Profit of winners + % Loss of losers)/(number of trades)

Average Bars Held - sum of bars in trades / number of trades

Winners:

Total Profit – Total profit generated by winning trades

Number of Winners – The number of winning trades

Average Profit - Average profit of the winning trades

Average Profit in Percentage – Average profit of the winning trades in percentage

Average Bars Held For Winners

Max Consecutive Winner

Largest Winner

Number of Bars in Largest Winner

Losers:

Total Loss – Total loss generated by loosing trades

Number of Losers – The number of loosing trades

Average Loss - Average loss of the loosing trades

Average Loss in Percentage – Average loss of the loosing trades in percentage

Average Bars Held For Losers

Max Consecutive Loser

Largest Loser

Number of Bars in Largest Loser

Drawdown:

Maximum Trade Drawdown - The largest peak to valley decline experienced in any single trade

Maximum Trade Drawdown In percentage - The largest peak to valley decline in percentage experienced in any single trade

Maximum system drawdown in dollar - The largest peak to valley decline experienced in portfolio equity

Maximum system drawdown in percentage - The largest peak to valley decline in percentage experienced in portfolio equity

Recovery factor - Net profit divided by Maximum system drawdown

CAR/MaxDD - Compound Annual percentage return divided by Maximum system percentage drawdown

RAR/MaxDD - Risk Adjusted Return divided by Maximum system percentage drawdown

Misc:

Number of trades – The total number of trades generated by the simulation

Percent of winners – The percentage of trade's winners

Annual turnover – Example: a value of 100% means that the average holding period of a trade is one year; a value of 200% means that the average holding period of a trade is six months

Total trading costs – The sum of all the trading costs generated by the simulation

Volume activity – The sum of all the positions size generated by the simulation

Percent trading cost of volume - Total trading costs / Volume activity

Percent positive days – The percentage of positive days in the simulation equity

Percent positive months - The percentage of positive months in the simulation equity

Percent positive years – The percentage of positive years in the simulation equity

Average daily return

Average weekly return

Average monthly return

Stats:

Ulcer index - Square root of sum of squared drawdown divided by number of bars

Ulcer performance index – This indicator is used to measure the riskiness of an investment

K Ratio - A ratio that is used in the performance evaluation of an equity relative to its risk. The ratio examines the consistency of equity's return over time. Higher is better

Standard deviation - A measure of the dispersion of a set of data from its mean. The more spread apart the data, the higher the deviation. Standard deviation is calculated as the square root of variance.

Downside standard deviation - The downside standard deviation from the ordinary standard deviation insofar as the sum is restricted to those returns that are less than the mean

Sharpe ratio - The Sharpe ratio tells us whether a portfolio's returns are due to smart investment decisions or a result of excess risk, the higher the better

Sortino ratio - The Sortino ratio is similar to the Sharpe ratio, except it uses downside deviation for the denominator instead of standard deviation, the higher the better

Profit factor - Profit of winners divided by loss of losers

Payoff ratio – Ratio of average wins / average loss

Performance Vs Benchmark:

Beta - A measure of the volatility, or systematic risk, of a security or a portfolio in comparison to the market as a whole; if a stock's beta is 1.2, it is theoretically 20% more volatile than the market.

Alpha - A measure of performance on a risk-adjusted basis, a positive alpha of 1.0 means the fund has outperformed its benchmark index by 1%. Correspondingly, a similar negative alpha would indicate an underperformance of 1%.

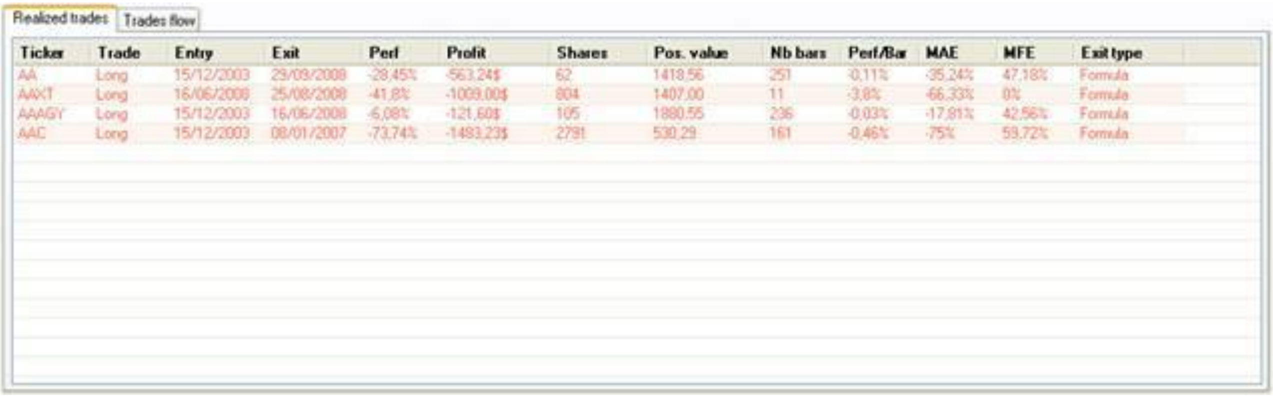
R Squared – The percentage, of a fund or security's movements, that can be explained by movements in a benchmark index. R-squared values range from zero to 100. An R-squared of 100 means that all movements of a security are completely explained by movements in the index

Correlation - a statistical measure of how the security and the benchmark move in relation to each other

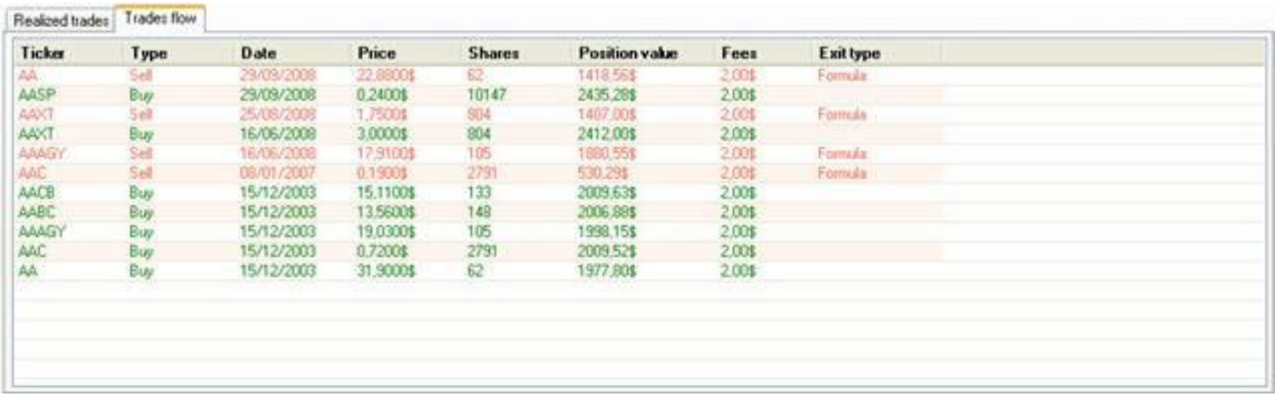
1.8.5 Trades

Realized trades: This window shows you the list of all realized trades.

Trades flow: This window shows you the list of all the trade's entries.



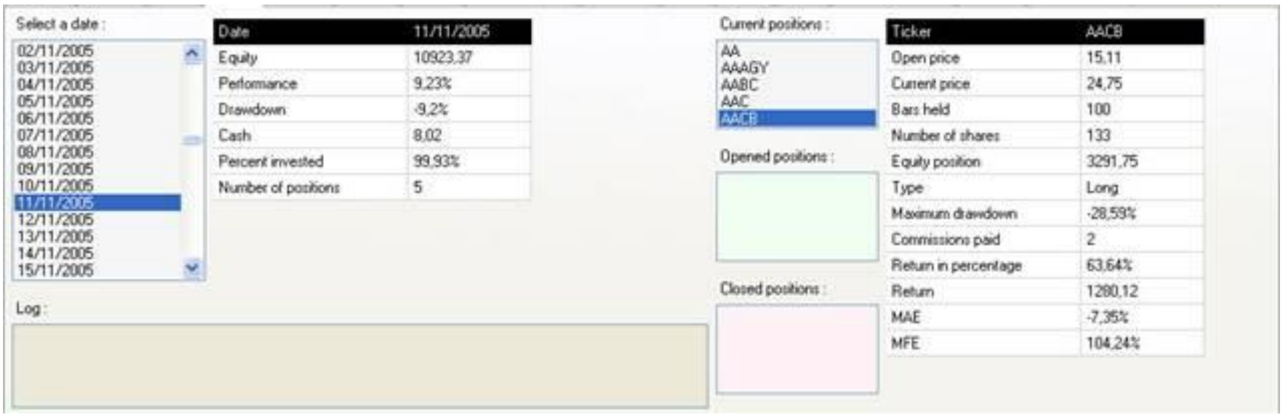
Ticker	Trade	Entry	Exit	Perf	Profit	Shares	Pos. value	Nb bars	Perf/Bar	MAE	MFE	Exit type
AA	Long	15/12/2003	29/09/2008	-28,45%	-563,24$	62	1418,56	251	-0,11%	-35,24%	47,18%	Formula
AAXT	Long	16/06/2008	25/08/2008	-41,8%	-1009,00$	804	1407,00	11	-3,8%	-66,33%	0%	Formula
AAAGY	Long	15/12/2003	16/06/2008	-6,08%	-121,60$	105	1880,55	236	-0,03%	-17,81%	42,56%	Formula
AAC	Long	15/12/2003	08/01/2007	-73,74%	-1483,23$	2791	530,29	161	-0,46%	-75%	59,72%	Formula



Ticker	Type	Date	Price	Shares	Position value	Fees	Exit type
AA	Sell	29/09/2008	22,8800$	62	1418,56$	2,00$	Formula
AASP	Buy	29/09/2008	0,2400$	10147	2435,28$	2,00$	
AAXT	Sell	25/08/2008	1,7500$	804	1407,00$	2,00$	Formula
AAXT	Buy	16/06/2008	3,0000$	804	2412,00$	2,00$	
AAAGY	Sell	16/06/2008	17,9100$	105	1880,55$	2,00$	Formula
AAC	Sell	08/01/2007	0,1900$	2791	530,29$	2,00$	Formula
AACB	Buy	15/12/2003	15,1100$	133	2009,63$	2,00$	
AABC	Buy	15/12/2003	13,5600$	148	2006,88$	2,00$	
AAAGY	Buy	15/12/2003	19,0300$	105	1998,15$	2,00$	
AAC	Buy	15/12/2003	0,7200$	2791	2009,52$	2,00$	
AA	Buy	15/12/2003	31,9000$	62	1977,80$	2,00$	

1.8.6 Detailed trades

This control shows you a snapshot of your portfolio for any date. It displays, for each date, the portfolio statistics, the current positions, the opened positions and the closed positions.
It also shows you the orders that were created for each date, and the orders that were rejected along with the reason of the rejection.



02/11/2005
03/11/2005
04/11/2005
05/11/2005
06/11/2005
07/11/2005
08/11/2005
09/11/2005
10/11/2005
11/11/2005
12/11/2005
13/11/2005
14/11/2005
15/11/2005

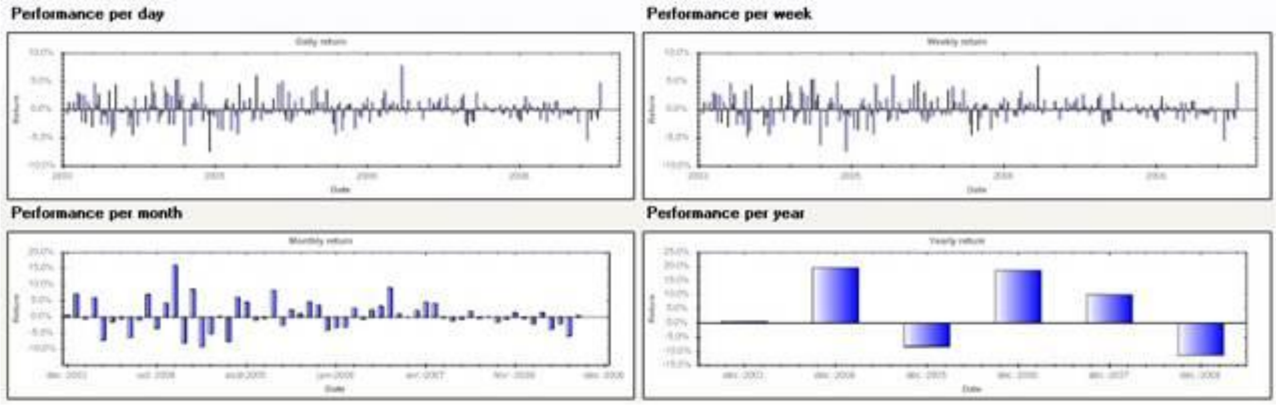
Date	11/11/2005
Equity	10923,37
Performance	9,23%
Drawdown	-9,2%
Cash	8,02
Percent invested	99,93%
Number of positions	5

AA
AAAGY
AABC
AAC
AACB

Ticker	AACB
Open price	15,11
Current price	24,75
Bars held	100
Number of shares	133
Equity position	3291,75
Type	Long
Maximum drawdown	-28,59%
Commissions paid	2
Return in percentage	63,64%
Return	1280,12
MAE	-7,35%
MFE	104,24%

1.8.7 Performance Per

This window shows you the performance of the trading system per day, week, month and year.



1.8.8 Symbols

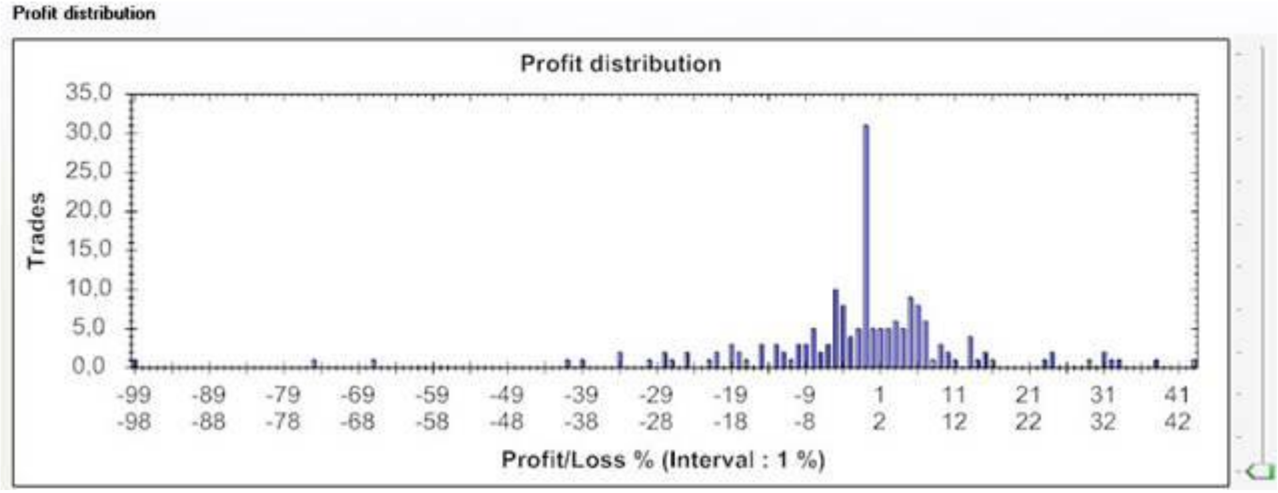
The Symbols tab contains the list of the symbols that were used by the simulation.

1.8.9 Formula

The formula tab contains the formula that was used to create the trading system.

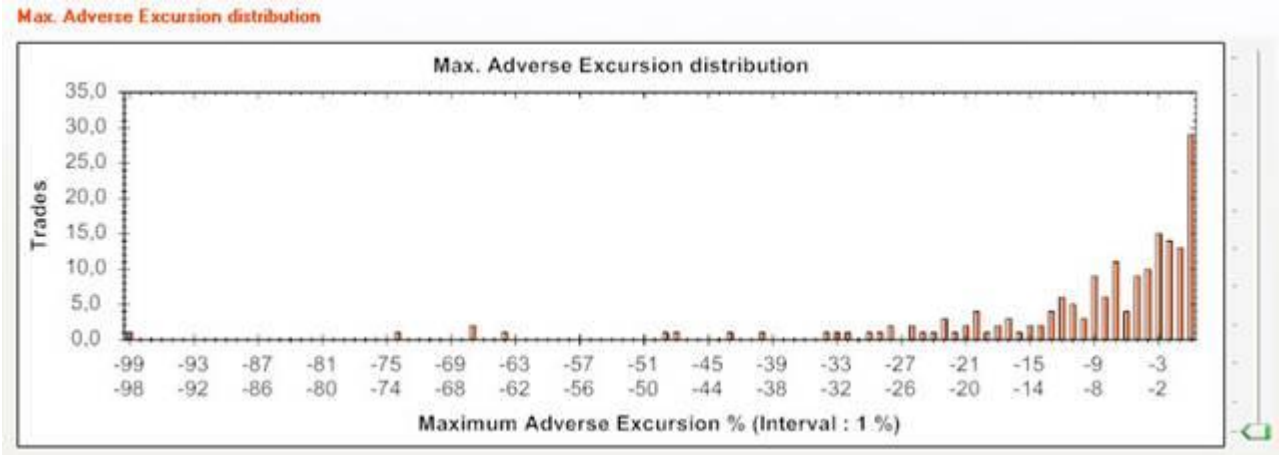
1.8.10 Profit distribution

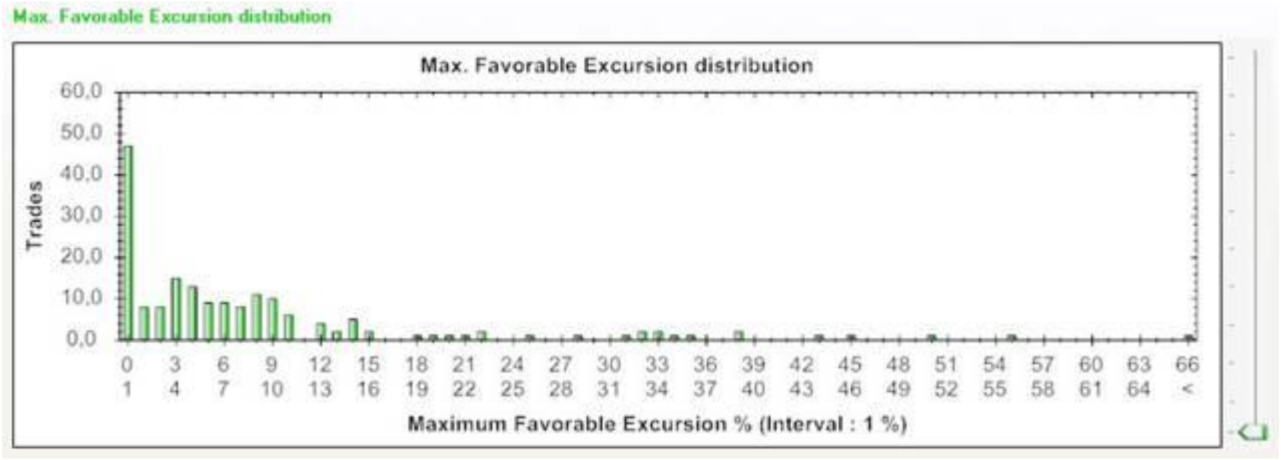
Display a graph that shows you the profit distribution.



1.8.11 MAE/MFE

Display two graphs, one for maximum adverse exclusion and the other for the maximum favorable exclusion.



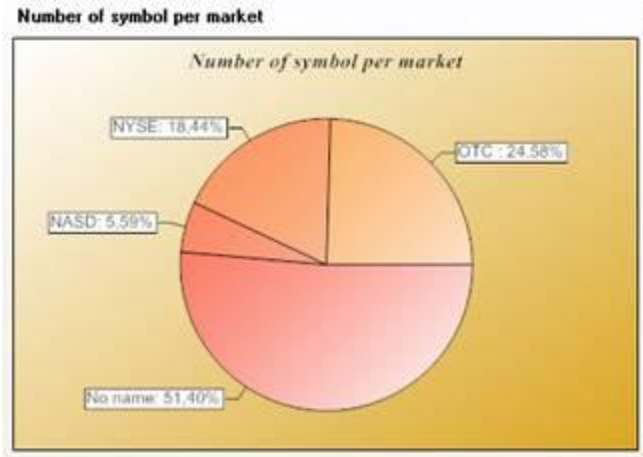


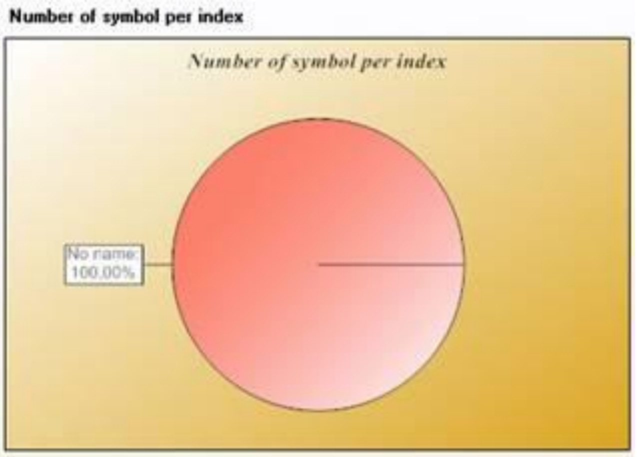
1.8.12 Custom graphs

Create custom graphs, could be Trade/MAE or Drawdown/MFE...

1.8.13 S.I.M.I

Shows you the position symbols distribution over markets, industries, sectors and indexes.



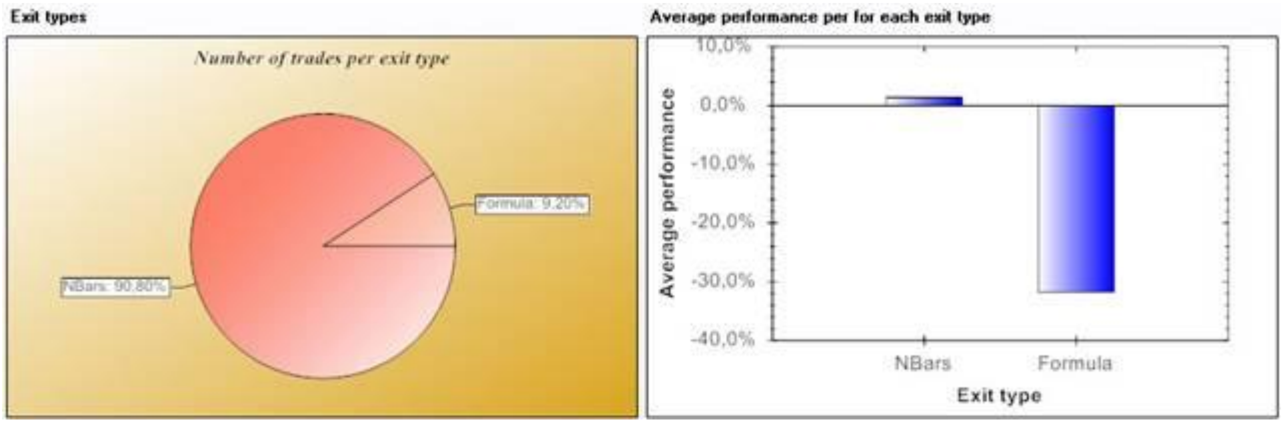


1.8.14 Reference symbol

The symbol used as reference.

1.8.15 Exit rules

Shows you the exit rules distribution and the average trade's performance for each exit rule.



1.8.16 Report settings

The Report settings tab allows you to define the default graph interval values.

1.8.17 Errors Tab

This tab displays the errors, if any, that occurred during the execution of the money management scripts.
The tab name will change to "No errors" if no error occurs during the money management scripts execution or if no money management rules are defined.

[TOP]

1.2.6.10.4 Advanced Money Management

1. Advanced Money Management

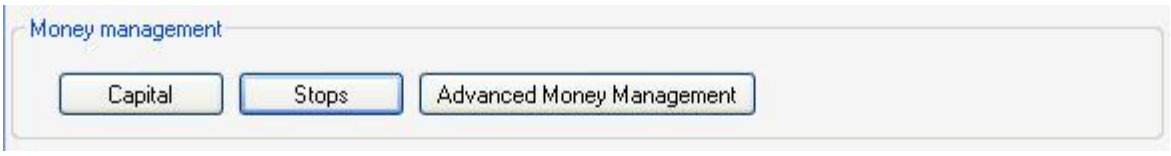
1.1. Introduction

The Money management plug-in let you take deep control over which instruments to buy, sell, short or cover, at which price and for which amount.
It will also let you you create advanced metrics to assess and evaluate any trading system.

You can also create variables within your advanced money management script and use the simulator optimizer to optimize these variables.

1.2. Create money management rules

In the 'Update trading system' or 'Create trading system' form, click on 'Advanced money management'; these money management rules you are about to create will be associated with the current trading system.



You can also create money management rules by selecting (*Analysis -> Advanced Money Management*)

1.3. Money management events

There are five money management events:

- OnNewPosition: Called before the application open a position
- OnClosePosition: Called before the application exit a position
- OnEndPeriod: Called at the end of a period (On historical back testing a period is a day, a week or a month, depending on the timeframe)
- OnStartSimulation: Called just before the simulator starts
- OnEndSimulation: Called just after the simulator ends

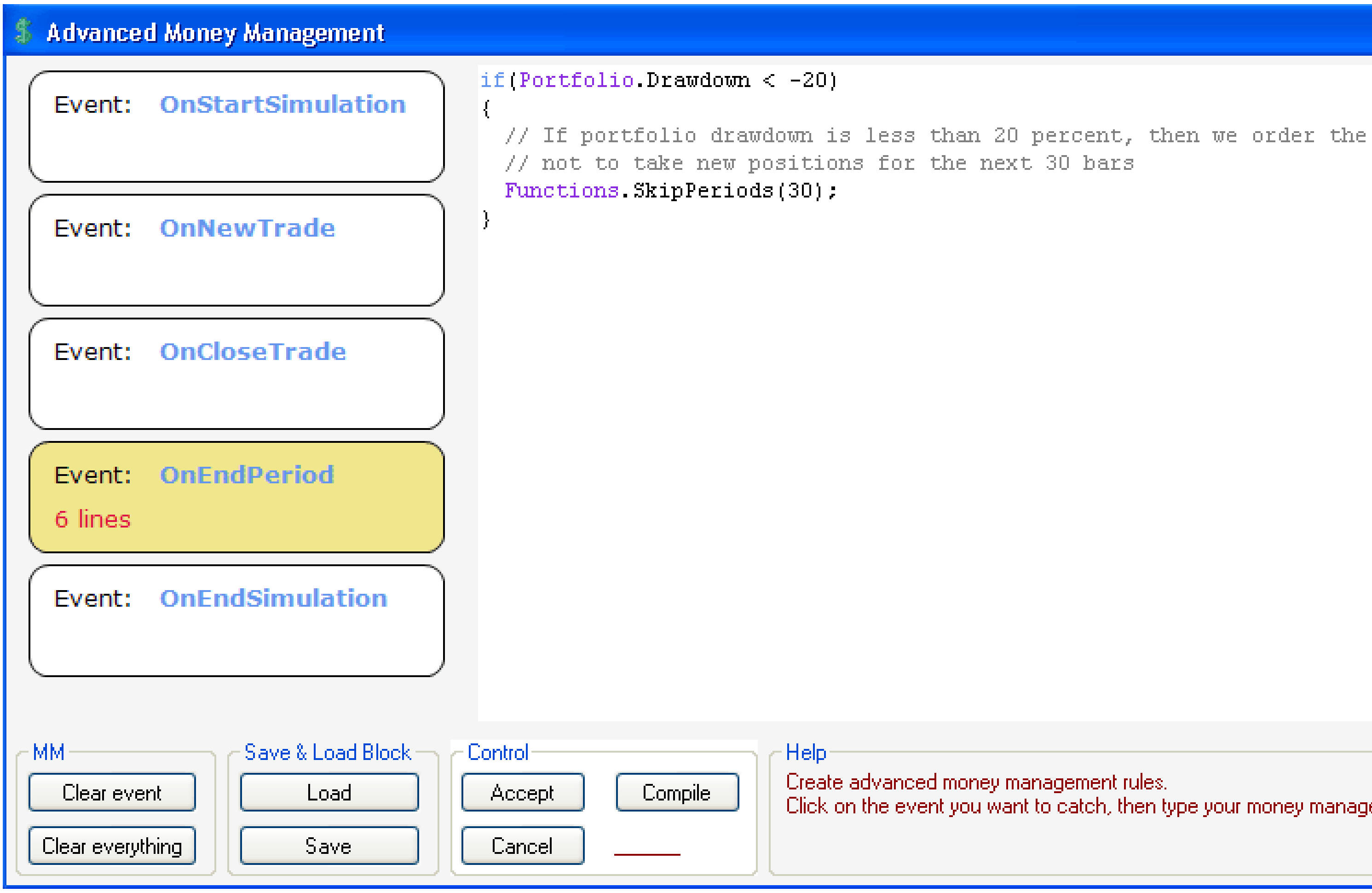
Each event has its own variables and functions; you can create a script for each event.

Example (OnEndPeriod Event):

We are going to create a script that will order the simulator not to take new positions for the next 30 bars if the portfolio drawdown is below -20 percent.

The Script:

```
if(Portfolio.Drawdown < -20)
{
  // If portfolio drawdown is less than 20 percent, then we order the simulator
  // not to take new positions for the next 30 bars
  Functions.SkipPeriods(30);
}
```



```
if(Portfolio.Drawdown < -20)
{
  // If portfolio drawdown is less than 20 percent, then we order the
  // not to take new positions for the next 30 bars
  Functions.SkipPeriods(30);
}
```

Create advanced money management rules.
Click on the event you want to catch, then type your money manag

Examples:

Reject an signal if the RSI value is lower than 50:

You can create directly this rule as a sell rule or you use the OnNewPosition event of the money management script.

```
TimeSeries t = Data.ParseFormula("a = rsi(14);").GetTimeSeries(NewPosition.Symbol
,"a");
if(t[0] < 50)
{
     Functions.RejectPosition();
}
```

Pair trading: In OnEndPeriod event

```
if(Your rule)
{
     MMPosition pos = Portfolio.GetPosition(symbol_1);
     if(pos != null && !pos.IsLong)
     {
          // Exit positions
          Functions.CloseAllPositions(0);
     }
     // Buy symbol_1 & Short symbol_2
     Functions.AddLongPosition(symbol_1, 0, Orders.OpenMarketOrder());
     Functions.AddShortPosition(symbol_2, 0, Orders.OpenMarketOrder());
}
else
{
    MMPosition pos = Portfolio.GetPosition(symbol_1);
    if(pos != null && pos.IsLong)
    {
        // Exit positions
        Functions.CloseAllPositions(0);
    }
    // Buy symbol_2 & Short symbol_1
    Functions.AddLongPosition(symbol_2, 0, Orders.OpenMarketOrder());
    Functions.AddShortPosition(symbol_1, 0, Orders.OpenMarketOrder());
}
```

1.2.6.10.5 Screener

1. Screener

1.1. Introduction

The screener plug-ins let you scan symbols quotes using defined criteria.

1.2. Columns Set

Analysis -> Columns Set

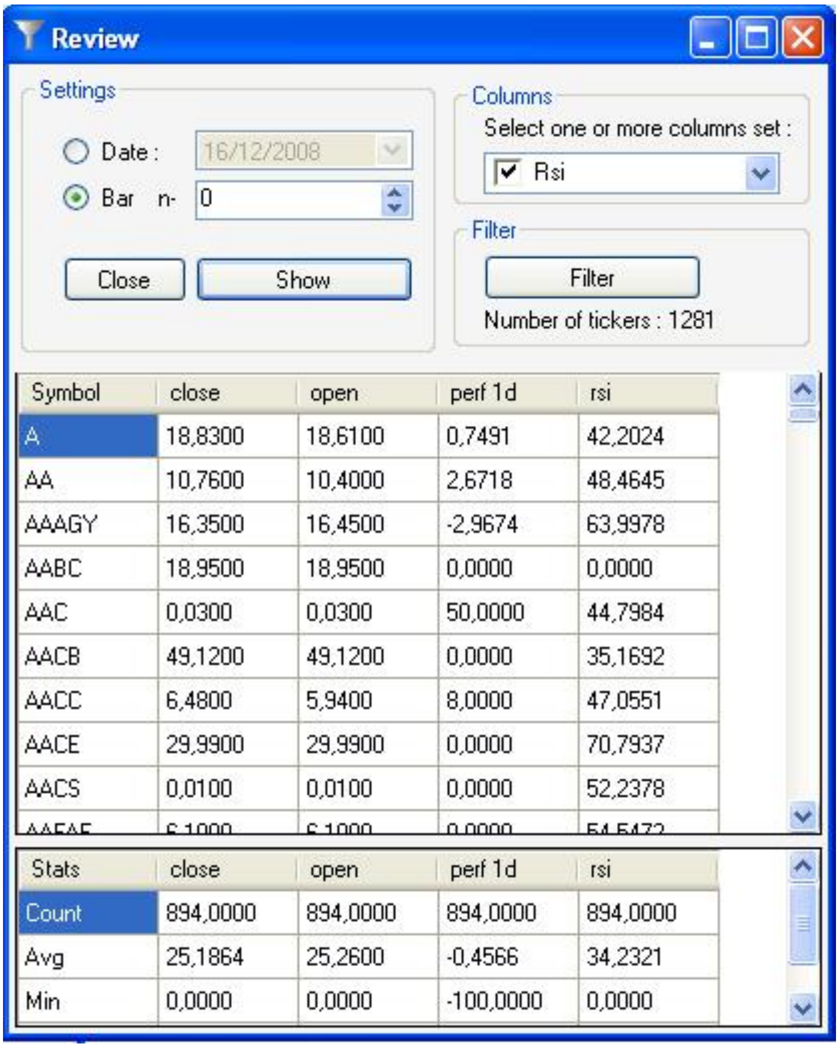
Example: the following formula will add a column whose name is 'Perf 1D' and will show "the performance of the symbol for one day" on the review or the screener report.

```
AddColumn ("Perf 1D", Perf (close, 1));
```

1.3. Create a Review

Analysis -> Review

Display symbols for the specified date or bar.



○ Date : 16/12/2008

● Bar n- 0

Select one or more columns set : ☑ Rsi

Number of tickers : 1281

Symbol	close	open	perf 1d	rsi
A	18,8300	18,6100	0,7491	42,2024
AA	10,7600	10,4000	2,6718	48,4645
AAAGY	16,3500	16,4500	-2,9674	63,9978
AABC	18,9500	18,9500	0,0000	0,0000
AAC	0,0300	0,0300	50,0000	44,7984
AACB	49,1200	49,1200	0,0000	35,1692
AACC	6,4800	5,9400	8,0000	47,0551
AACE	29,9900	29,9900	0,0000	70,7937
AACS	0,0100	0,0100	0,0000	52,2378
AAEAE	6,1000	6,1000	0,0000	54,5472

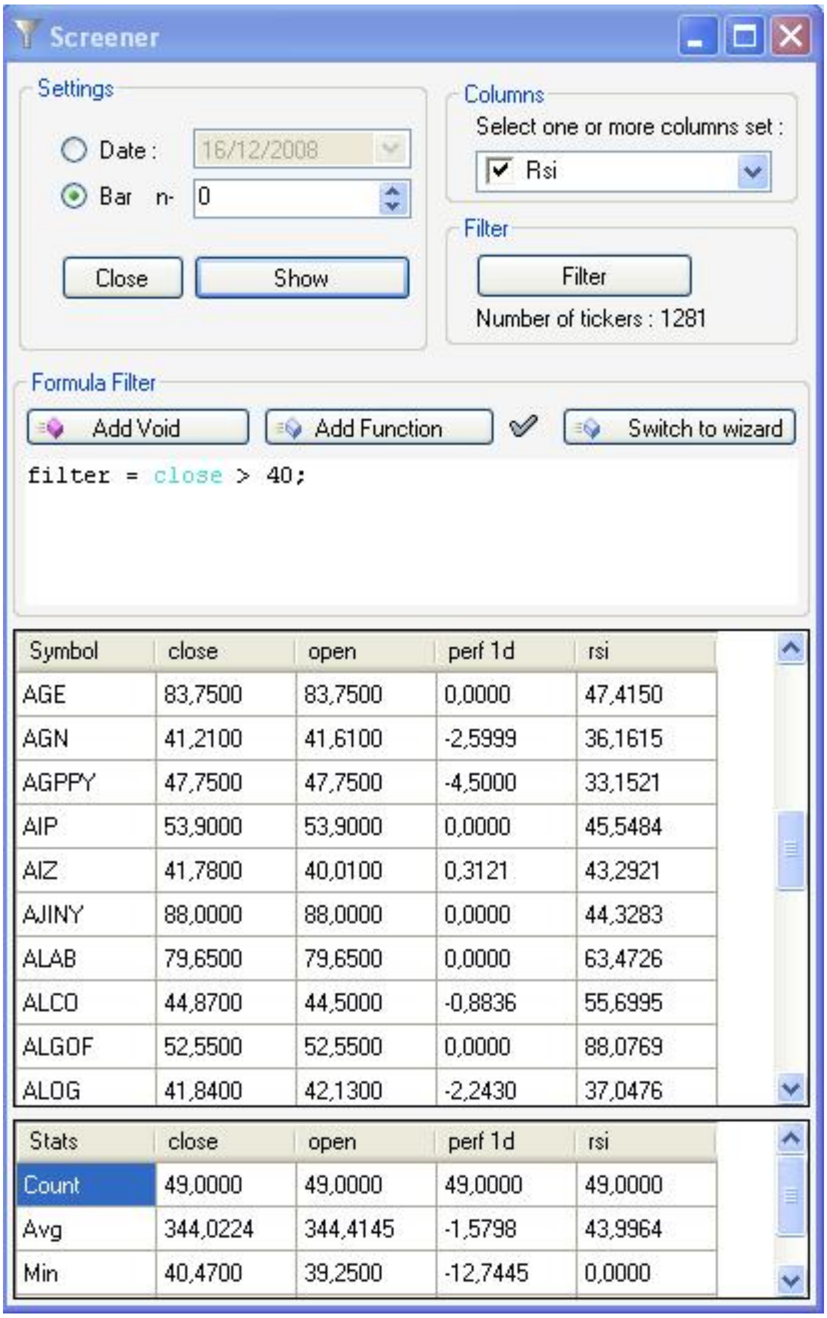
Stats	close	open	perf 1d	rsi
Count	894,0000	894,0000	894,0000	894,0000
Avg	25,1864	25,2600	-0,4566	34,2321
Min	0,0000	0,0000	-100,0000	0,0000

1.4. Create a Screener

Analysis -> Screener

The screener scans the symbol's quotes and displays the symbols that meet your criteria.
Criteria are controlled by the variable 'filter'.

Example:



○ Date : 16/12/2008
● Bar n- 0

Select one or more columns set :
☑ Rsi

Number of tickers : 1281

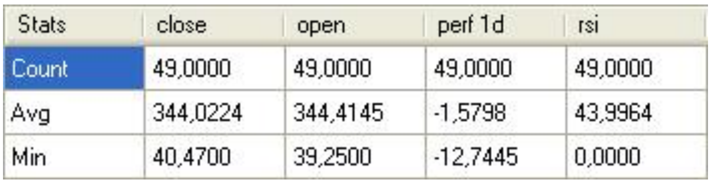
```
filter = close > 40;
```

Symbol	close	open	perf 1d	rsi
AGE	83,7500	83,7500	0,0000	47,4150
AGN	41,2100	41,6100	-2,5999	36,1615
AGPPY	47,7500	47,7500	-4,5000	33,1521
AIP	53,9000	53,9000	0,0000	45,5484
AIZ	41,7800	40,0100	0,3121	43,2921
AJINY	88,0000	88,0000	0,0000	44,3283
ALAB	79,6500	79,6500	0,0000	63,4726
ALCO	44,8700	44,5000	-0,8836	55,6995
ALGOF	52,5500	52,5500	0,0000	88,0769
ALOG	41,8400	42,1300	-2,2430	37,0476

Stats	close	open	perf 1d	rsi
Count	49,0000	49,0000	49,0000	49,0000
Avg	344,0224	344,4145	-1,5798	43,9964
Min	40,4700	39,2500	-12,7445	0,0000

1.4.1. Stats

Stats grid, in the review or the screener form, takes each column and calculate the maximum, minimum and average values.



Stats	close	open	perf 1d	rsi
Count	49,0000	49,0000	49,0000	49,0000
Avg	344,0224	344,4145	-1,5798	43,9964
Min	40,4700	39,2500	-12,7445	0,0000

[TOP]

1.2.6.11 Artificial Intelligence

1.2.6.11.1 Artificial Intelligence

1. Artificial Intelligence

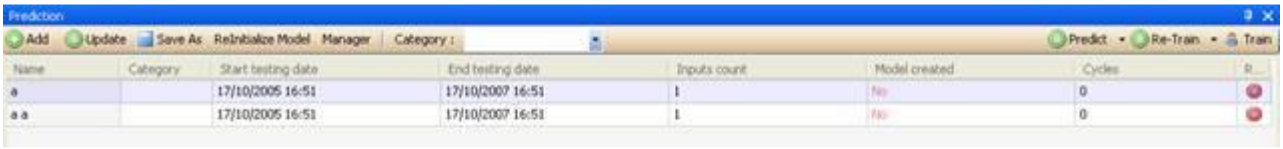
Artificial intelligence is a plug-in that let you build prediction model using neural network and SVM (Support vector machine) systems.

1.1. Neural Networks

http://en.wikipedia.org/wiki/Neural_network

1.2. Create a prediction model

To create a prediction model, open the 'prediction model' form (Click on 'AI' then 'Prediction'), in the new form, click on 'Add'.
The prediction model has many settings; those settings will help you fine-tune your model.

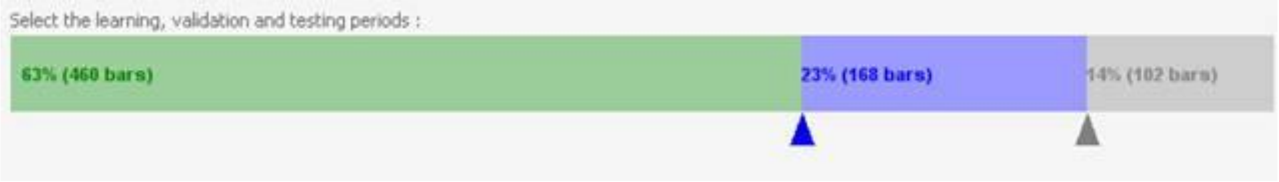


Name	Category	Start testing date	End testing date	Inputs count	Model created	Cycles	R...
a		17/10/2005 16:51	17/10/2007 16:51	1	No	0	●
a a		17/10/2005 16:51	17/10/2007 16:51	1	No	0	●

1.3. Learning, validation and testing periods

Each prediction model must have three periods.

The first period or the learning period is used to train the model.
The second period or the validation period is used to choose the best model among all the generated models.
The third period or the testing period is used to test the model with new data. (Data on which the model was not trained)
Note that the model is trained only with data from the learning period.


Select the learning, validation and testing periods :

To change the learning, validation and testing periods, click on the triangles and move them.

1.4. Learning and validation samples

There are two options:

Normal: The first bars (depending on the learning period) will be associated to the learning samples, and then the next bars will be associated with the validation samples.

Random: The learning and validation periods will be merged and the learning and validation samples will be chosen randomly.

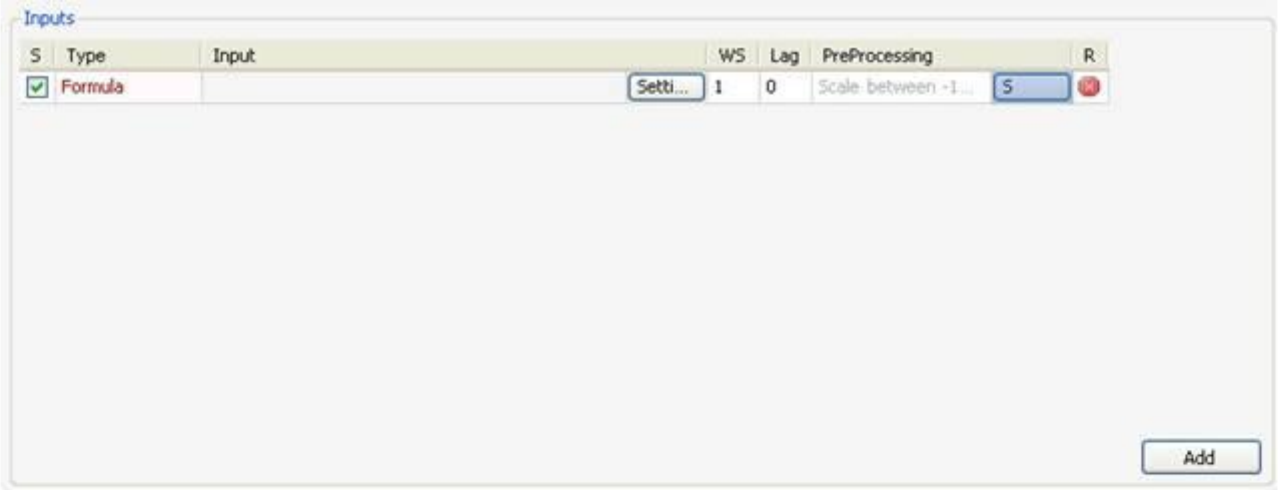
1.5. Periodicity

This option let you choose the period that the prediction item will use.

1.6. Inputs

Inputs are time-series that will be used to train the prediction item.
This is probably the most important step in the creation of the prediction item.

Add a new input by clicking on 'Add'.



Inputs

S	Type	Input		WS	Lag	PreProcessing		R
☑	Formula		Setti...	1	0	Scale between -1...	S	✖

Columns:

- Select: The select checkbox indicate whether to enable or disable the current input
- Input Type: Select the input type you would like to use.
- Input: Display information about the input time-series.
- Settings: Define settings related to the input you have selected

- WS: Let you create multiple inputs from this input.
  Example: if you create 'close' as input and set WS to two, the system will create two time-series. The first one is 'close' and the second one is 'close one bar ago' or 'ref (close, 1)'.
- Lag: Let you specify the lag to include for the input.
  Example: if you create 'close' as input and set Lag to 3, the system will create the following input 'close three bar ago' or 'ref(close, 3)'.
- Preprocessing: Let you select the pre-processing calculation that will be applied to the input.

- R: This button is used to remove the selected input.

Input Types:

- Symbol Field: You can choose between open, high, low, close or volume times-series.
- Symbol Return: This creates a formula that returns the performance of one of these time-series: open, high, low, close, volume.
- Formula: This lets you build your own time-series using the formula editor.
- Extern Symbol: This lets you create a time-series based on an external symbol data.
- Database Field: This lets you create a time-series from a field database.

1.7. Output

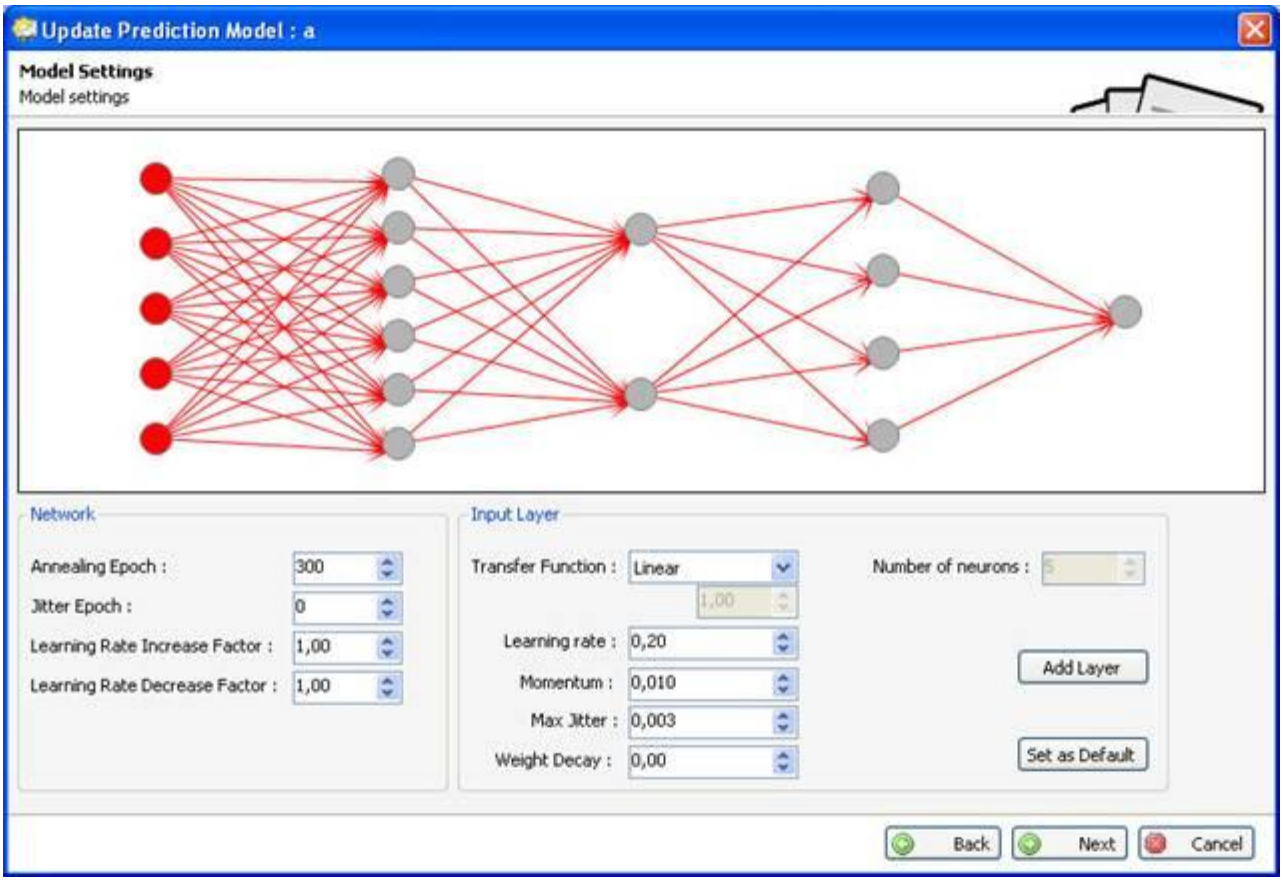
The output is the time-series that will be predicted.
You can predict a symbol price, a symbol returns or any other time-series.

The output has the same settings as inputs.

The 'WS' column doesn't exists in the output settings.
You can create only one output.

1.8. Neural network model settings

A neural network model has different settings that can dramatically improve or reduce the performance of the prediction model.



Update Prediction Model : a

Model Settings
Model settings

Network

Annealing Epoch : 300
Jitter Epoch : 0
Learning Rate Increase Factor : 1,00
Learning Rate Decrease Factor : 1,00

Input Layer

Transfer Function : Linear
1,00
Number of neurons : 5
Learning rate : 0,20
Momentum : 0,010
Max Jitter : 0,003
Weight Decay : 0,00

1.9. Network

A neural network model is composed of one input layer, one output layer and zero, one or many hidden layers.
Click on the circles to select a layer.

Network settings:

- Annealing Epoch: The epoch at which the learning rate increase or decrease is performed
- Jitter Epoch: The epoch at which the Jitter is performed. Jitter is the process of adding small random noise to weights of all synapses at regular intervals in order to get the network out of a potential local minimum
- Learning Rate Increase Factor: Let the network learn faster in order to improve the global error, value should lies between 1.00 and 1.03

- Learning Rate Decrease Factor: Let the network learn slower in order to improve the global error, value should lies between 0.97 and 1.00

Input layer:
The input layer is the first layer of the neural network model.
The number of inputs determines the number of nodes for this layer.

Hidden layers:
Hidden layers are layers that are always between the input and the output layer.

Output layer:
The output layer is the last layer of the neural network model.
There is only one node in the output layer.
The value that leaves this layer is the predicted value.

Layer settings:

- Transfer function: Activation or transfer function in a back propagation network defines the way to obtain output of a neuron given the collective input from source synapses.
- Learning rate: Learning rate is one of the parameters that govern how fast a neural network learns and how effective the training is.
- Momentum: Momentum term associated with each synapse represents its tendency to retain its previous weight change. This factor varies from zero to one.
- Max Jitter: The maximum limit to the random noise added to weights while Jitter, Jitter is the process of adding small random noise to weights of all synapses at regular intervals in order to get the network out of a potential local minimum.
- Weight Decay: Each time the neural network is trained, the weights of synapses are multiplied by this 'Weight Decay'. Weight decay value is usually between 0.9 and 1.

1.10. Filter

The filter lets you create a formula that will be used to reject certain bars from the learning process.
Example: If you want to reject bars where the volume was insignificant then type something like this: 'volume < 100'.

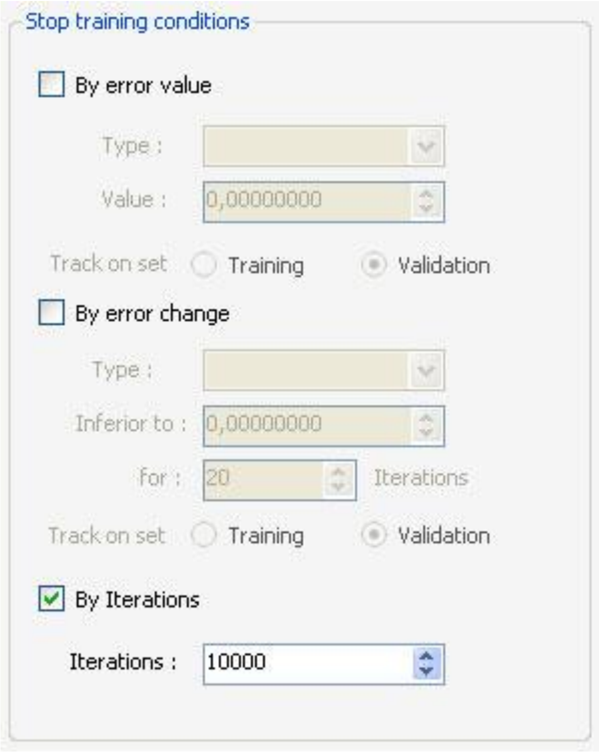
1.11. Symbols

Select the symbols that will be used in the learning process.

1.12. Stop training settings

Specify when to stop the training, there are three options:

- By error value: Stop when the error value reaches a certain level.
- By error change: Stop when the error change goes below a certain value for a determined number of iterations.
- By iterations: Stop after a specific number of iterations.



Stop training conditions

□ By error value

Type :

Value : 0,00000000

Track on set ○ Training ● Validation

□ By error change

Type :

Inferior to : 0,00000000

for : 20 Iterations

Track on set ○ Training ● Validation

☑ By Iterations

Iterations : 10000

1.13. Selecting the best model

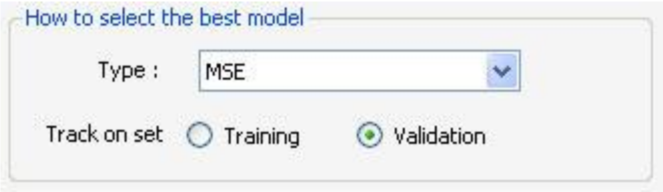
Select the model that will be used in prediction among all the models created during the training.

Type: Select the model based on one of these values:

Network Error:

- MSE (Root mean square error)
- POCID (Directional accuracy)
- POCD (Prediction sign accuracy)
- UTHEIL (Coefficient U de Theil)

Track on set: Choose whether to select the best model among the training models or the validation models.



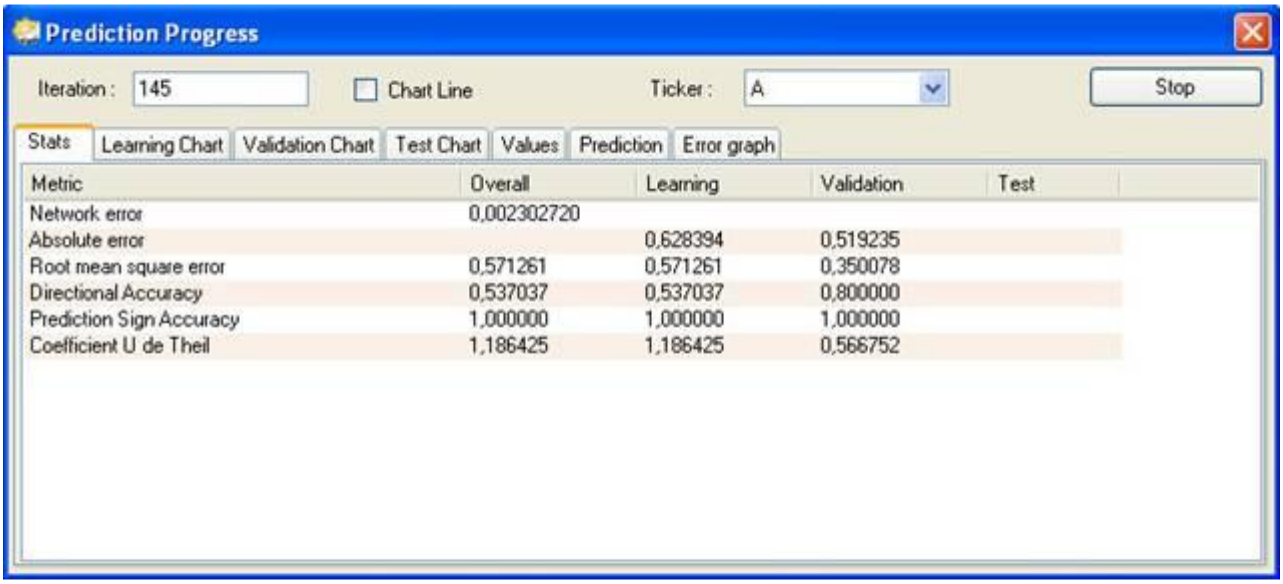
How to select the best model

Type : MSE

Track on set ○ Training ● Validation

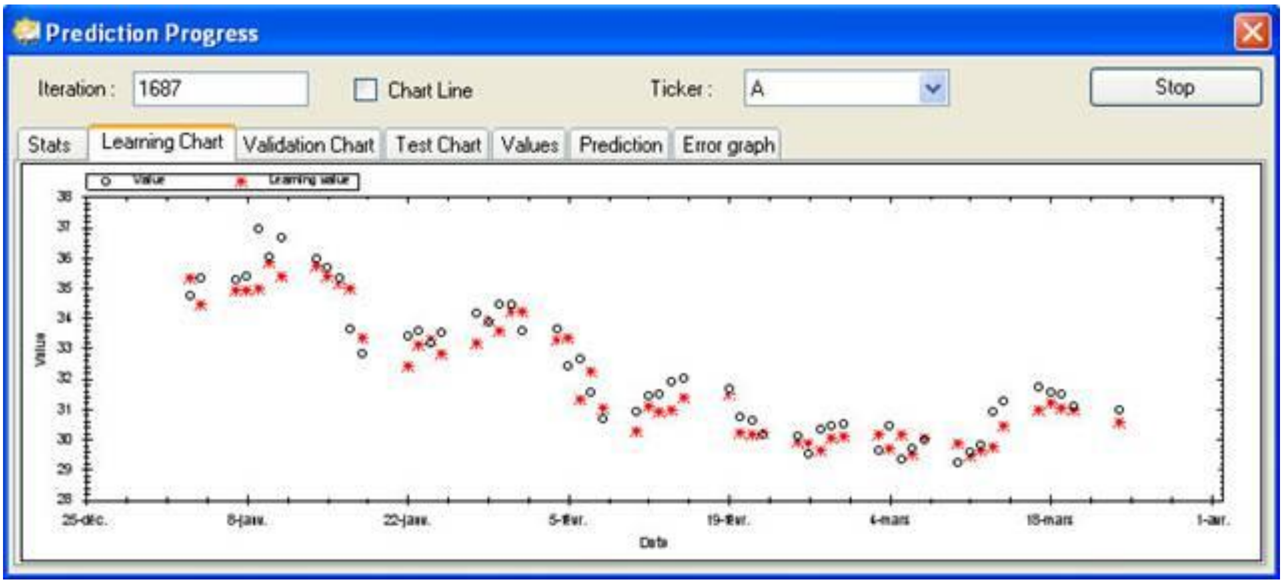
1.14. Train a model

In the 'Prediction' form, select an item then click on 'Train', the 'Prediction Progress' form appears.

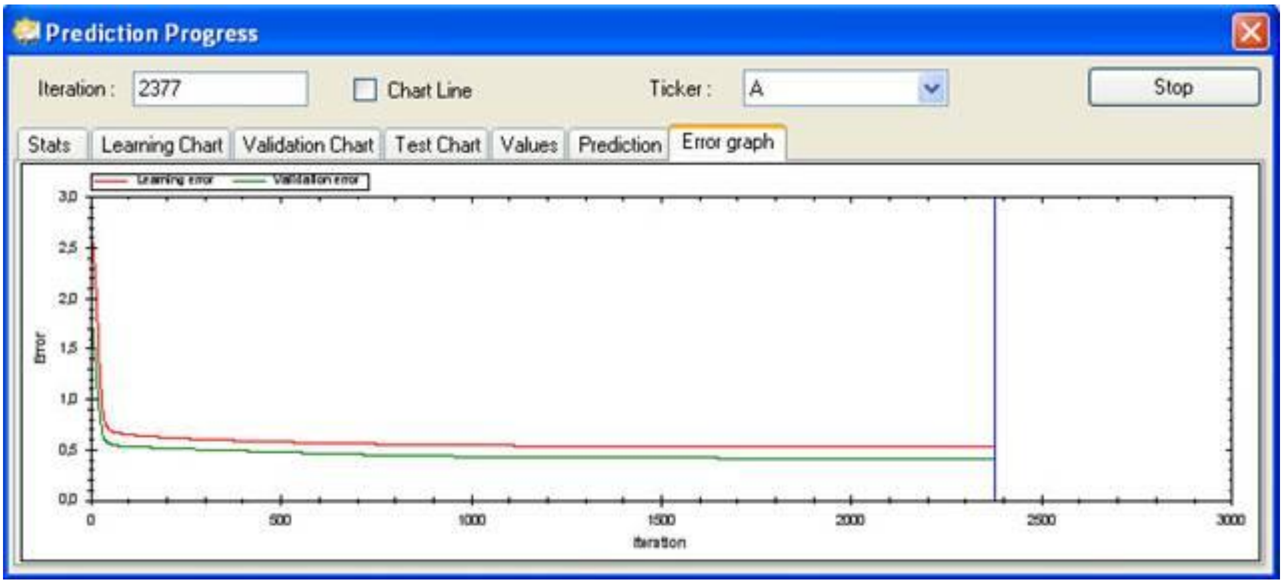


Iteration : 145 ☐ Chart Line Ticker : A

Metric	Overall	Learning	Validation	Test
Network error	0,002302720			
Absolute error		0,628394	0,519235	
Root mean square error	0,571261	0,571261	0,350078	
Directional Accuracy	0,537037	0,537037	0,800000	
Prediction Sign Accuracy	1,000000	1,000000	1,000000	
Coefficient U de Theil	1,186425	1,186425	0,566752	



Iteration : 1687 ☐ Chart Line Ticker : A



Iteration : 2377 ☐ Chart Line Ticker : A

The blue line refers to the iteration that produced the best prediction model depending on your settings.

1.15. Prediction

In the 'Prediction' form, select an item then click on the small arrow next to 'Predict'.
You have the choice to choose between 'Predict' and 'Predict a value'.

1.15.1. Predict

This option gives you the ability to predict values for a range of dates.

1.15.2. Predict a value

This option gives you the ability to predict a value for a specific date.

1.16. Re-Training

You have the possibility to re-train a prediction model on new data. To do so, click on 'Re-Train' button in the 'Prediction' form.

1.16.1. Re-Training Settings

Re-Training Settings lets you choose the range of dates and the learning and validation samples that will be used for the re-training.

1.17. Update a model

In the 'Prediction' form, select an item then click on 'Update'.
You will lose the data of the prediction model if you update the latter.

1.18. Re-initialize a model

Click on 'Reinitialize Model' in the 'Prediction' form to delete the date of the model.

1.19. Access a model from a formula

Two functions let you access a model data from a formula:

```
Predict("category name", "model name") ;
```

Return a time-series that contains the prediction values.

```
PredictSymbol("category name", "model name") ;
```

Return '1' if the prediction model was trained using the current symbol, otherwise returns '0'.

[TOP]

1.2.6.11.2 Optimizer

1. Optimizer

1.1. Introduction

The optimizer is a plug-in that uses artificial intelligence algorithms to optimize rules, ranking systems, trading systems and prediction items.

1.2. Optimizer algorithms

The plug-in provides two algorithms:

- Genetic algorithm
- Population-based incremental learning

Example:
Take a list of rules with 1000 rules.
For these rules, we have more than billions of combination which will take years to back-test.
Optimization algorithms are used to find good solutions among all the combinations in a short period.

1.2.1. Genetic algorithm

These are the genetic algorithm parameters:

- Number of generation
- Population size
- Selection method: Elite, Rank or Roulette
- Crossover rate
- Mutation rate
- Random selection portion

http://en.wikipedia.org/wiki/Genetic_algorithm

1.2.2. PBIL algorithm

These are the PBIL algorithm parameters:

- Number of generation
- Population size
- Learning rate
- Number of best solutions to use in learning

http://en.wikipedia.org/wiki/Population-based_incremental_learning

1.3. Optimize items

1.3.1. Optimize list of rules

Search for the best combination of rules among a list of rules.

- In the 'Optimizer' form, click on 'Create'
- Select the optimization algorithm and 'List of Rules' in the 'What do you want to optimize?" list box
- Click on 'Next'

- Update the algorithm settings
- Click on 'Next'
- Select the number of rules
- Select a list of rules, by clicking on 'Select rules'.
- For list of rules, select whether to always include the current rule or not by checking or un-checking 'Always visible' checkbox
- Select the output
- Define a fitness formula
- Click on 'Next'
- Select the number of items to keep in the report
- Select the symbols the optimizer will use
- Click on 'Next'
- The optimizer item will now appears in the optimize manager.

Example:
List of rules contains 100 rules, from rule1 to rule100.

The optimizer will try to find good combination of rules.
The result will look like this:

Formula	Fitness
Rule87 && rule3	0.9
Rule12 && rule2	0.76
Rule33	0.66
Rule67 && Rule32 && Rule2	0.5

Fitness could be the return of the symbol or something else; it depends on the 'Fitness' formula.

1.3.2. Optimize ranking system

Try to find which combination of nodes and nodes' weight gives the best fitness score for the ranking system.

- In the 'Optimizer' form, click on 'Create'
- Select the optimization algorithm and 'Ranking System' in the 'What do you want to optimize?" list box
- Click on 'Next'
- Update the algorithm settings
- Click on 'Next'
- Select a ranking system by clicking on 'Select a ranking system'
- Select the number of buckets and the rebalance period
- Choose whether to optimize nodes' weight by checking or un-checking 'Optimize node weights' checkbox
- Type a fixed rule if you want to use a fixed rule in the ranking system evaluation
- Select the output
- Define a fitness formula
- Click on 'Next'
- Select the number of items to keep in the report
- Select the symbols the optimizer will use
- Click on 'Next'
- The optimizer item will now appears in the optimize manager.

1.3.3. Optimize trading system

Try to find which trading system works best given a buy, sell, short and cover list of rules, money management variables, and others settings.

- In the 'Optimizer' form, click on 'Create'
- Select the optimization algorithm and 'Trading System' in the 'What do you want to optimize?" list box
- Click on 'Next'
- Update the algorithm settings
- Click on 'Next'
- Select the number of buy and sell rules
- For each buy and sell rule, select a list of rules by clicking on 'Select rules'
- For each buy and sell rule, select whether to always include this rule or not by checking or un-checking 'Always visible' checkbox
- Define a fitness formula
- On the 'Settings' tab, type a fixed buy rule and a fixed sell rule if you want to include fixed rules in your trading system
- Select 'Optimization', to include capital and stops optimizable variables
- Click on 'Next'
- Select the number of items to keep in the report
- Select the symbols the optimizer will use
- Click on 'Next'
- The optimizer item will now appears in the optimize manager.

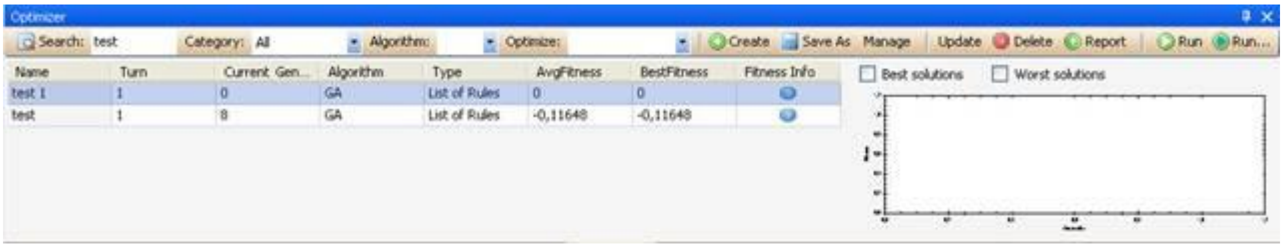
1.3.4. Optimize prediction item

Try to find which prediction item leads to a better prediction accuracy given a list of inputs.

- In the 'Optimizer' form, click on 'Create'
- Select the optimization algorithm and 'Prediction Model' in the 'What do you want to optimize?" list box
- Click on 'Next'
- Update the algorithm settings
- Click on 'Next'
- Select a prediction item then click on 'Load Inputs'
- Update the prediction item if necessary by clicking on 'Update Item'
- Select the input that will always be used in iterations by checking them (Fixed inputs)
- Define a fitness formula
- Click on 'Next'
- Select the number of items to keep in the report
- Select the symbols the optimizer will use
- Click on 'Next'
- The optimizer item will now appears in the optimize manager.

1.4. Optimize manager

The 'Optimize manager' list all optimize items (*AI -> Optimizer*)



Name	Turn	Current Gen...	Algorithm	Type	AvgFitness	BestFitness	Fitness Info
test 1	1	0	GA	List of Rules	0	0	●
test	1	8	GA	List of Rules	-0,11648	-0,11648	●

To create an optimize item, click on 'Create'.
To remove an optimize item, click on 'Delete'.
To update an optimize item, click on 'Update'.

To load the optimizer report, click on 'Report'.
To start an optimize item, click either on 'Run' or on 'Run...'

Running an optimize item indefinitely, using 'Run...', means that when the optimizer reach the maximum number of generation or the fitness value converge, the software re-run the optimize item and increase the value of 'Turn' by one.

1.5. Turn

If you run an optimize item multiple times, you may end up each time with different results.
This is because the optimizer algorithms do not look for all combinations but instead try to find good solutions in a short period.
This is why we introduced the concept of 'Turn'.
The optimizer report includes the best items for all the turns (not only the last turn)..

[TOP]

1.2.6.12 External

1.2.6.12.1 Portfolio123

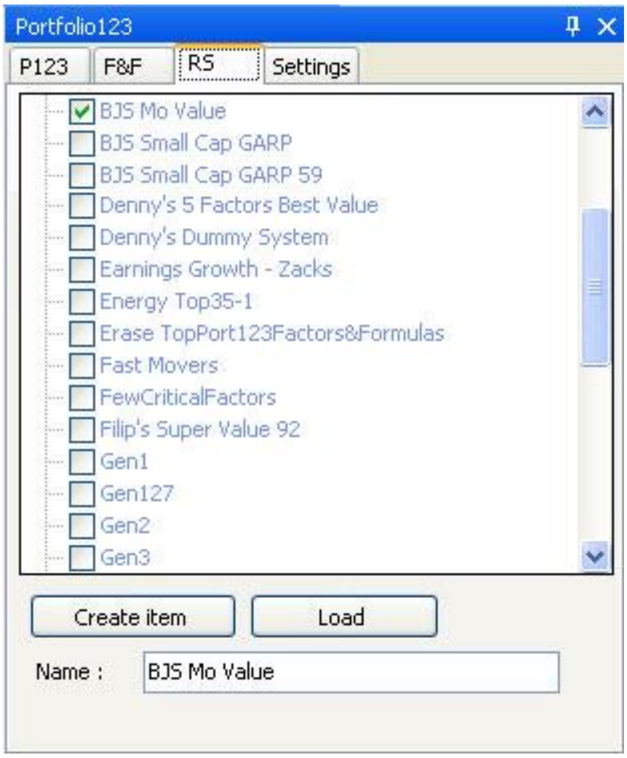
1. Portfolio123

1.1. Introduction

Portfolio123 is a service that lets you create ranking system models based on fundamental and technical items. Porfolio123 database contains more than 4000 fundamental ratios for all US companies. For more info, visit their website on www.portfolio123.com.
This plug-in lets Portfolio123 subscribers import factors, formulas and ranking system data.

1.2. Import a ranking system

- Open the Portfolio123 plug-in (*Extern->Portfolio123*)
- Select 'Settings' tab, make sure you have inserted your portfolio123 login and password.
- Select 'RS' tab
- Click on 'Load' if no ranking systems appear
- Select a ranking system
- Type in a name in the 'Name' text box
- Click on 'Create Item'
- Switch to 'P123' tab, your newly created item appears there.



Name : BJS Mo Value

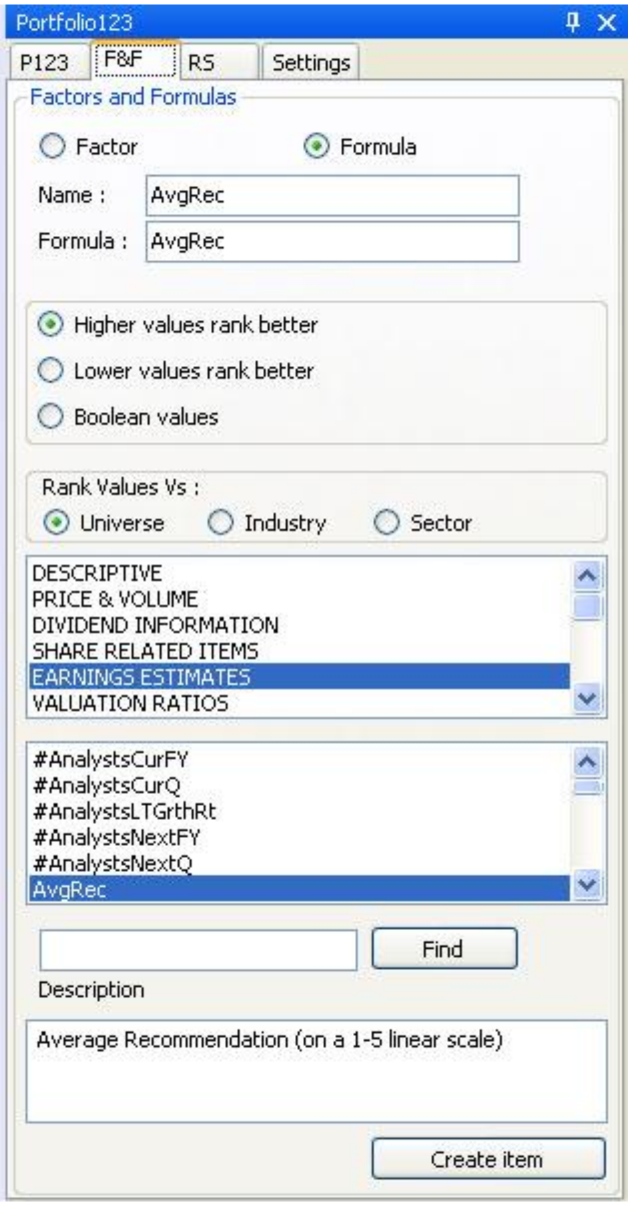
1.3. Create a factor

- Open the Portfolio123 plug-in (*Extern->Portfolio123*)
- Select 'F&F' tab
- Check the radio box 'Factor'
- Select a factor (select a category then a factor)

- Set the factor settings (lower or higher, rank vs.)
- Click on 'Create Item'
- Switch to 'P123' tab, your newly create item appears there.

1.4. Create a function

- Open the Portfolio123 plug-in (*Extern->Portfolio123*)
- Select 'F&F' tab
- Check the radio box 'Formula'
- Type a formula then type the formula name
- Set the formula settings (lower or higher, rank vs.)
- Click on 'Create Item'
- Switch to 'P123' tab, your newly create item appears there.



Portfolio123

Factors and Formulas

○ Factor ● Formula

Name : AvgRec

Formula : AvgRec

● Higher values rank better
○ Lower values rank better
○ Boolean values

Rank Values Vs :
● Universe ○ Industry ○ Sector

DESCRIPTIVE
PRICE & VOLUME
DIVIDEND INFORMATION
SHARE RELATED ITEMS
EARNINGS ESTIMATES
VALUATION RATIOS

#AnalystsCurFY
#AnalystsCurQ
#AnalystsLTGrthRt
#AnalystsNextFY
#AnalystsNextQ
AvgRec

Description

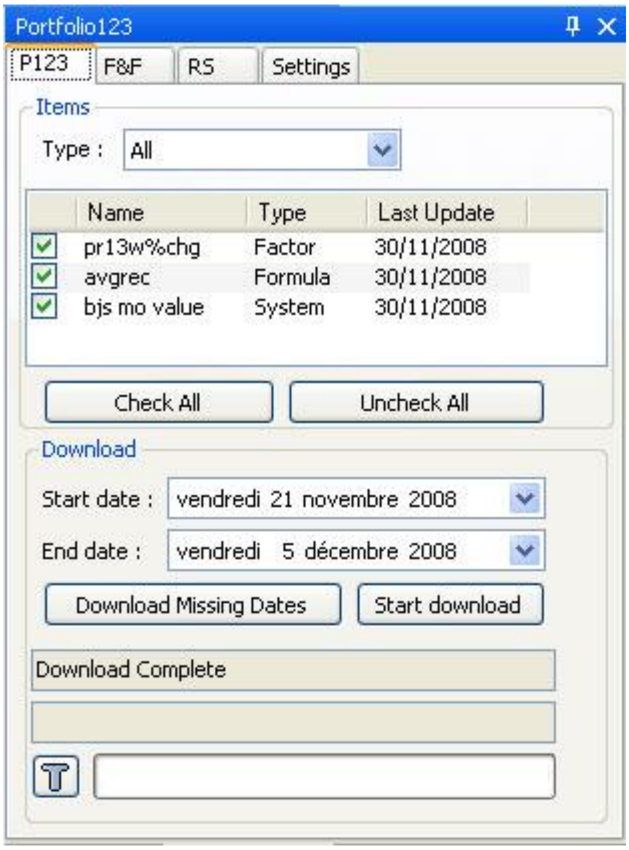
Average Recommendation (on a 1-5 linear scale)

1.5. Download Portfolio123 data

- Open the Portfolio123 plug-in (*Extern->Portfolio123*)
- Select 'P123' tab
- Check the items you want to download

- Select the start date and the end date
- Click on 'Start download'

If you want to start the download from the last update date, click on 'Download Missing Dates' instead of 'Start download'.



Type : All

	Name	Type	Last Update
☑	pr13w%chg	Factor	30/11/2008
☑	avgrec	Formula	30/11/2008
☑	bjs mo value	System	30/11/2008

Start date : vendredi 21 novembre 2008

End date : vendredi 5 décembre 2008

Download Complete

[TOP]

1.3 QuantShare Language

1.3.1 Date-Time

1.3.1.1 Year

YEAR
Year

Date-time

SYNTAX	**Year()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the current bar's year
ADDITIONAL INFO	
EXAMPLE	Year()

[TOP]

1.3.1.2 Date

DATE
Date

Date-time

SYNTAX	**Date()**
RETURNS	TEXT ARRAY
DESCRIPTION	Returns the bar's Date
ADDITIONAL INFO	
EXAMPLE	Date()

[TOP]

1.3.1.3 DateTicks

DATETICKS
Date Ticks

Date-time

SYNTAX	**DateTicks()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the number of ticks that represent the current date and time
ADDITIONAL INFO	
EXAMPLE	DateTicks()

[TOP]

1.3.1.4 Day

DAY
Day

Date-time

SYNTAX	**Day()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the current bar's day of the month
ADDITIONAL INFO	
EXAMPLE	Day()

[TOP]

1.3.1.5 DayOfWeek

DAYOFWEEK
Day of Week

Date-time

SYNTAX	**DayOfWeek()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the current bar's day of week (1 for Monday, 2 for Tuesday, 3 for Wednesday...)
ADDITIONAL INFO	
EXAMPLE	DayOfWeek()

[TOP]

1.3.1.6 DayOfYear

DAYOFYEAR
Day of Year

Date-time

SYNTAX	**DayOfYear()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the current bar's day of the year
ADDITIONAL INFO	
EXAMPLE	DayOfYear()

[TOP]

1.3.1.7 Hour

HOUR
Hour

Date-time

SYNTAX	**Hour()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the current bar's hour
ADDITIONAL INFO	
EXAMPLE	Hour()

[TOP]

1.3.1.8 Interval

INTERVAL
Interval in ticks

Date-time

SYNTAX	**Interval()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the number of ticks between the current and previous bar date
ADDITIONAL INFO	
EXAMPLE	Interval()

[TOP]

1.3.1.9 Minute

MINUTE
Minute

Date-time

SYNTAX	**Minute()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the current bar's minute
ADDITIONAL INFO	
EXAMPLE	Minute()

[TOP]

1.3.1.10 Month

MONTH
Month

Date-time

SYNTAX	**Month()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the current bar's month
ADDITIONAL INFO	
EXAMPLE	Month()

[TOP]

1.3.1.11 NbDays

NBDAYS
Number of days

Date-time

SYNTAX	**NbDays(ARRAY period)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the number of days between the current bar and the bar referenced by the period value
ADDITIONAL INFO	
EXAMPLE	NbDays(5)

[TOP]

1.3.1.12 Now

NOW
Now

Date-time

SYNTAX	**Now()**
RETURNS	TEXT ARRAY
DESCRIPTION	Returns the current Date
ADDITIONAL INFO	
EXAMPLE	Now()

[TOP]

1.3.1.13 Second

SECOND
Second

Date-time

SYNTAX	**Second()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the current bar's second
ADDITIONAL INFO	
EXAMPLE	Second()

[TOP]

1.3.1.14 TimeTicks

TIMETICKS
Time Ticks

Date-time

SYNTAX	**TimeTicks()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the number of ticks that represent the current time
ADDITIONAL INFO	
EXAMPLE	TimeTicks()

[TOP]

1.3.1.15 Week

WEEK
Week

Date-time

SYNTAX	**Week()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the current bar's week
ADDITIONAL INFO	
EXAMPLE	Week()

[TOP]

1.3.2 Application Info

1.3.2.1 NbGroups

NBGROUPS
Number of groups

Application Info

SYNTAX	**NbGroups()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the number of groups
ADDITIONAL INFO	
EXAMPLE	NbGroups()

[TOP]

1.3.2.2 NbIndexes

NBINDEXES
Number of indices

SYNTAX	**NbIndexes()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the number of indexes
ADDITIONAL INFO	
EXAMPLE	NbIndexes()

[TOP]

1.3.2.3 NbIndustries

NBINDUSTRIES
Number of industries

SYNTAX	**NbIndustries()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the number of industries
ADDITIONAL INFO	
EXAMPLE	NbIndustries()

[TOP]

1.3.2.4 NbInGroup

NBINGROUP
Number of symbols in the specified group

Application Info

SYNTAX	**NbInGroup(ARRAY group name)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the number of symbols that are included in the specified group
ADDITIONAL INFO	
EXAMPLE	NbInGroup("Stock")

1.3.2.5 NbInIndex

NBININDEX
Number of symbols in the specified index

SYNTAX	**NbInIndex(ARRAY index name)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the number of symbols that are included in the specified index
ADDITIONAL INFO	
EXAMPLE	NbInIndex("SP500")

[TOP]

1.3.2.6 NbInIndustry

NBININDUSTRY
Number of symbols in the specified industry

Application Info

SYNTAX	**NbInIndustry(ARRAY industry name)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the number of symbols that are included in the specified industry
ADDITIONAL INFO	
EXAMPLE	NbInIndustry("Drug Stores")

[TOP]

1.3.2.7 NbInMarket

NBINMARKET
Number of symbols in the specified market

Application Info

SYNTAX	**NbInMarket(ARRAY market name)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the number of symbols that are included in the specified market
ADDITIONAL INFO	
EXAMPLE	NbInMarket("NASDQ")

[TOP]

1.3.2.8 NbInSector

NBINSECTOR
Number of symbols in the specified sector

Application Info

SYNTAX	**NbInSector(ARRAY sector name)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the number of symbols that are included in the specified sector
ADDITIONAL INFO	
EXAMPLE	NbInSector("Services")

[TOP]

1.3.2.9 NbMarkets

NBMARKETS
Number of markets

Application Info

SYNTAX	**NbMarkets()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the number of markets
ADDITIONAL INFO	
EXAMPLE	NbMarkets()

[TOP]

1.3.2.10 NbSectors

NBSECTORS
Number of sectors

SYNTAX	**NbSectors()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the number of sectors
ADDITIONAL INFO	
EXAMPLE	NbSectors()

[TOP]

1.3.3 Candlestick Pattern

1.3.3.1 Cdl2crows 0

CDL2CROWS
Two Crows

Candlestick Pattern

SYNTAX **Cdl2crows()**

RETURNS NUMERIC ARRAY

DESCRIPTION Two Crows, The Two Crows Pattern is a 3-day pattern.

ADDITIONAL INFO
Signal: Bearish
Pattern: Reversal
Reliability: Medium
During an uptrend we see the market closing lower after an opening gap. Then we see a black day that fills the gap creating the Bearish Two Crows Pattern. It suggests the erosion of the uptrend, and warns about a possible trend reversal.

EXAMPLE Cdl2crows()

[TOP]

1.3.3.2 Cdl2crows 1

CDL2CROWS
Two Crows

Candlestick Pattern

SYNTAX **Cdl2crows(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Two Crows, The Two Crows Pattern is a 3-day pattern.

ADDITIONAL INFO
Signal: Bearish
Pattern: Reversal
Reliability: Medium
During an uptrend we see the market closing lower after an opening gap. Then we see a black day that fills the gap creating the Bearish Two Crows Pattern. It suggests the erosion of the uptrend, and warns about a possible trend reversal.

EXAMPLE Cdl2crows(open, high, low, close)

[TOP]

1.3.3.3 Cdl3blackcrows 0

CDL3BLACKCROWS
Three Black Crows

Candlestick Pattern

SYNTAX	**Cdl3blackcrows()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Three Black Crows, The Three Black Crows got their name from the resemblance of three crows looking down from their perch from a tree.
ADDITIONAL INFO	Signal: Bearish Pattern: reversal Reliability: high Three long black days with each successive open being within the body of the previous day and each successive close being below the previous day's and near the day's low.
EXAMPLE	Cdl3blackcrows()

[TOP]

1.3.3.4 Cdl3blackcrows 1

CDL3BLACKCROWS
Three Black Crows

Candlestick Pattern

SYNTAX **Cdl3blackcrows(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Three Black Crows, The Three Black Crows got their name from the resemblance of three crows looking down from their perch from a tree.

ADDITIONAL INFO
Signal: Bearish
Pattern: reversal
Reliability: high
Three long black days with each successive open being within the body of the previous day and each successive close being below the previous day's and near the day's low.

EXAMPLE Cdl3blackcrows(open, high, low, close)

[TOP]

1.3.3.5 Cdl3inside 0

CDL3INSIDE
Three Inside Up/Down

Candlestick Pattern

SYNTAX **Cdl3inside()**

RETURNS NUMERIC ARRAY

DESCRIPTION Three Inside Up/Down, Note that after the long candle day that is in the same direction of the trend that the Harami pattern occurs.

ADDITIONAL INFO
Signal: Bullish
Pattern: reversal
Reliability: high
A bullish Harami pattern is followed by a white day that has a higher close than the second day.

EXAMPLE Cdl3inside()

[TOP]

1.3.3.6 Cdl3inside 1

CDL3INSIDE
Three Inside Up/Down

Candlestick Pattern

SYNTAX	**Cdl3inside(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Three Inside Up/Down, Note that after the long candle day that is in the same direction of the trend that the Harami pattern occurs.
ADDITIONAL INFO	Signal: Bullish Pattern: reversal Reliability: high A bullish Harami pattern is followed by a white day that has a higher close than the second day.
EXAMPLE	Cdl3inside(open, high, low, close)

[TOP]

1.3.3.7 Cdl3linestrike 0

CDL3LINESTRIKE
Three-Line Strike

Candlestick Pattern

SYNTAX **Cdl3linestrike()**

RETURNS NUMERIC ARRAY

DESCRIPTION Three-Line Strike : This function is contained within the Pattern Recognition set of indicators.

ADDITIONAL INFO
Three Line Strike Bullish
Signal: Bullish
Pattern: continuation
Reliability: low
Identification
Three long white days with consecutively higher closes are followed by a fourth day that gaps open in the direction of the trend and closes below the open of the first day.
Three Line Strike Bearish
Signal: Bearish
Pattern: continuation
Reliability: low
Identification
Three long black days with consecutively lower closes is followed by a fourth day that gaps in the direction of the trend and closes above the open of the first day.

EXAMPLE Cdl3linestrike()

[TOP]

1.3.3.8 Cdl3linestrike 1

CDL3LINESTRIKE
Three-Line Strike

Candlestick Pattern

SYNTAX **Cdl3linestrike(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Three-Line Strike : This function is contained within the Pattern Recognition set of indicators.

ADDITIONAL INFO
Three Line Strike Bullish
Signal: Bullish
Pattern: continuation
Reliability: low
Identification
Three long white days with consecutively higher closes are followed by a fourth day that gaps open in the direction of the trend and closes below the open of the first day.
Three Line Strike Bearish
Signal: Bearish
Pattern: continuation
Reliability: low
Identification
Three long black days with consecutively lower closes is followed by a fourth day that gaps in the direction of the trend and closes above the open of the first day.

EXAMPLE Cdl3linestrike(open, high, low, close)

[TOP]

1.3.3.9 Cdl3outside 0

CDL3OUTSIDE
Three Outside Up/Down

Candlestick Pattern

SYNTAX	**Cdl3outside()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Three Outside Up/Down : This function is contained within the Pattern Recognition set of indicators.
ADDITIONAL INFO	Three Outside Down Signal: Bearish Pattern: reversal Reliability: high Identification A bearish Engulfing pattern is followed by a black day whose close is lower than the second day. Three Outside Up Signal: Bullish Pattern: reversal Reliability: high Identification A bullish Engulfing pattern is followed by a white day whose close is higher than the second day.
EXAMPLE	Cdl3outside()

[TOP]

1.3.3.10 Cdl3outside 1

CDL3OUTSIDE
Three Outside Up/Down

Candlestick Pattern

SYNTAX	**Cdl3outside(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Three Outside Up/Down : This function is contained within the Pattern Recognition set of indicators.
ADDITIONAL INFO	Three Outside Down Signal: Bearish Pattern: reversal Reliability: high Identification A bearish Engulfing pattern is followed by a black day whose close is lower than the second day. Three Outside Up Signal: Bullish Pattern: reversal Reliability: high Identification A bullish Engulfing pattern is followed by a white day whose close is higher than the second day.
EXAMPLE	Cdl3outside(open, high, low, close)

[TOP]

1.3.3.11 Cdl3staRsinsouth 0

CDL3STARSINSOUTH
Three Stars In The South

Candlestick Pattern

SYNTAX **Cdl3staRsinsouth()**

RETURNS NUMERIC ARRAY

DESCRIPTION Three Stars In The South, The slow down of the trend is visually obvious.

ADDITIONAL INFO
Signal: Bullish
Pattern: reversal
Reliability: moderate
A long black day with a long lower shadow is followed by a similar but smaller black day whose lower shadow is shallower than the first day. The third day is a small Black Marubozu (open is the high of the day and the close is the low of the day) that lies within the second dayï¿½s trading range.

EXAMPLE Cdl3staRsinsouth()

[TOP]

1.3.3.12 Cdl3staRsinsouth 1

CDL3STARSINSOUTH
Three Stars In The South

Candlestick Pattern

SYNTAX **Cdl3staRsinsouth(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Three Stars In The South, The slow down of the trend is visually obvious.

ADDITIONAL INFO Signal: Bullish
Pattern: reversal
Reliability: moderate
A long black day with a long lower shadow is followed by a similar but smaller black day whose lower shadow is shallower than the first day. The third day is a small Black Marubozu (open is the high of the day and the close is the low of the day) that lies within the second dayï¿½s trading range.

EXAMPLE Cdl3staRsinsouth(open, high, low, close)

[TOP]

1.3.3.13 Cdl3whitesoldiers 0

CDL3WHITESOLDIERS
Three Advancing White Soldiers

Candlestick Pattern

SYNTAX **Cdl3whitesoldiers()**

RETURNS NUMERIC ARRAY

DESCRIPTION Three Advancing White Soldiers, The Three White Soldiers (also known as The Advancing Three White Soldiers) is a healthy market reversal pattern.

ADDITIONAL INFO
Signal: Bullish
Pattern: reversal
Reliability: high
Three long white days with each successive open being within the body of the previous day and each successive close being higher than the previous day and near the dayï¿½s high.

EXAMPLE Cdl3whitesoldiers()

[TOP]

1.3.3.14 Cdl3whitesoldiers 1

CDL3WHITESOLDIERS
Three Advancing White Soldiers

Candlestick Pattern

SYNTAX	**Cdl3whitesoldiers(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Three Advancing White Soldiers, The Three White Soldiers (also known as The Advancing Three White Soldiers) is a healthy market reversal pattern.
ADDITIONAL INFO	Signal: Bullish Pattern: reversal Reliability: high Three long white days with each successive open being within the body of the previous day and each successive close being higher than the previous day and near the dayï¿½s high.
EXAMPLE	Cdl3whitesoldiers(open, high, low, close)

[TOP]

1.3.3.15 CdlAbandonedbaby 0

CDLABANDONEDBABY
Abandoned Baby

Candlestick Pattern

SYNTAX **CdlAbandonedbaby(ARRAY penetration)**

RETURNS NUMERIC ARRAY

DESCRIPTION Abandoned Baby, A rare reversal pattern characterized by a gap followed by a Doji, which is then followed by another gap in the opposite direction.

ADDITIONAL INFO
Abandoned Baby Bearish
Signal: Bearish
Pattern: reversal
Reliability: high
Identification
A long white day is followed by a Doji that gaps in the direction of the trend. Then a black day occurs gapping in the opposite direction with no overlapping shadows.
Abandoned Baby Bullish
Signal: Bullish
Pattern: reversal
Reliability: high
Identification
A long black day is followed by a Doji that gaps in the direction of the trend. Then a white day occurs gapping in the opposite direction with no overlapping shadows.

EXAMPLE CdlAbandonedbaby(14)

[TOP]

1.3.3.16 CdlAbandonedbaby 1

CDLABANDONEDBABY
Abandoned Baby

Candlestick Pattern

SYNTAX **CdlAbandonedbaby(ARRAY open, ARRAY high, ARRAY low, ARRAY close, ARRAY penetration)**

RETURNS NUMERIC ARRAY

DESCRIPTION Abandoned Baby, A rare reversal pattern characterized by a gap followed by a Doji, which is then followed by another gap in the opposite direction.

ADDITIONAL INFO
Abandoned Baby Bearish
Signal: Bearish
Pattern: reversal
Reliability: high
Identification
A long white day is followed by a Doji that gaps in the direction of the trend. Then a black day occurs gapping in the opposite direction with no overlapping shadows.
Abandoned Baby Bullish
Signal: Bullish
Pattern: reversal
Reliability: high
Identification
A long black day is followed by a Doji that gaps in the direction of the trend. Then a white day occurs gapping in the opposite direction with no overlapping shadows.

EXAMPLE CdlAbandonedbaby(open, high, low, close, 14)

[TOP]

1.3.3.17 CdlAdvanceblock 0

CDLADVANCEBLOCK
Advance Block

Candlestick Pattern

SYNTAX **CdlAdvanceblock()**

RETURNS NUMERIC ARRAY

DESCRIPTION Advance Block, The Advance Block is somewhat indicative as the Three White Soldiers but it is a bearish signal.

ADDITIONAL INFO
Signal: Bearish
Pattern: reversal
Reliability: moderate
Three white days. Each successive day opens within the body of the previous day and closes above the previous day. The bodies of the candles get progressively smaller with the upper shadows of day 2 and 3 getting progressively longer.

EXAMPLE CdlAdvanceblock()

[TOP]

1.3.3.18 CdlAdvanceblock 1

CDLADVANCEBLOCK
Advance Block

Candlestick Pattern

SYNTAX	**CdlAdvanceblock(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Advance Block, The Advance Block is somewhat indicative as the Three White Soldiers but it is a bearish signal.
ADDITIONAL INFO	Signal: Bearish Pattern: reversal Reliability: moderate Three white days. Each successive day opens within the body of the previous day and closes above the previous day. The bodies of the candles get progressively smaller with the upper shadows of day 2 and 3 getting progressively longer.
EXAMPLE	CdlAdvanceblock(open, high, low, close)

[TOP]

1.3.3.19 CdlBelthold 0

CDLBELTHOLD
Belt-hold

Candlestick Pattern

SYNTAX	**CdlBelthold()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Belt-hold, The Belt Hold lines are formed by single candlesticks.
ADDITIONAL INFO	Belt Hold Bearish Signal: Bearish Pattern: reversal Reliability: low Identification A black day occurs with no upper shadow and a close near the day's low. Belt Hold Bullish Signal: Bullish Pattern: reversal Reliability: low Identification A white day occurs with no lower shadow and a close near the dayï¿½s high.
EXAMPLE	CdlBelthold()

[TOP]

1.3.3.20 CdlBelthold 1

CDLBELTHOLD
Belt-hold

Candlestick Pattern

SYNTAX **CdlBelthold(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Belt-hold, The Belt Hold lines are formed by single candlesticks.

ADDITIONAL INFO
Belt Hold Bearish
Signal: Bearish
Pattern: reversal
Reliability: low
Identification
A black day occurs with no upper shadow and a close near the day's low.
Belt Hold Bullish
Signal: Bullish
Pattern: reversal
Reliability: low
Identification
A white day occurs with no lower shadow and a close near the dayï¿½s high.

EXAMPLE CdlBelthold(open, high, low, close)

[TOP]

1.3.3.21 CdlBreakaway 0

CDLBREAKAWAY
Breakaway

Candlestick Pattern

SYNTAX **CdlBreakaway()**

RETURNS NUMERIC ARRAY

DESCRIPTION Breakaway, If a trend has been evident, the breakaway pattern, whether bullish or bearish initially indicates the acceleration of that trend.

ADDITIONAL INFO
Bullish Breakaway
Signal: Bullish
Pattern: Reversal
Reliability: Medium
Identification
There is a downtrend but we also see that the prices bottom out and level off now. The result is a long white candlestick that however does not close the initial downward gap of the first and second days. This suggests a short-term reversal.
Bearish Breakaway
Signal: Bearish
Pattern: Reversal
Reliability: Medium
Identification
We see this pattern during an uptrend marked with a bullish surge that eventually weakens. This weakening is illustrated by a long black candlestick that is unable to close the gap into the body of the first day. These events warn us about a short-term reversal.

EXAMPLE CdlBreakaway()

[TOP]

1.3.3.22 CdlBreakaway 1

CDLBREAKAWAY
Breakaway

Candlestick Pattern

SYNTAX **CdlBreakaway(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Breakaway, If a trend has been evident, the breakaway pattern, whether bullish or bearish initially indicates the acceleration of that trend.

ADDITIONAL INFO
Bullish Breakaway
Signal: Bullish
Pattern: Reversal
Reliability: Medium
Identification
There is a downtrend but we also see that the prices bottom out and level off now. The result is a long white candlestick that however does not close the initial downward gap of the first and second days. This suggests a short-term reversal.
Bearish Breakaway
Signal: Bearish
Pattern: Reversal
Reliability: Medium
Identification
We see this pattern during an uptrend marked with a bullish surge that eventually weakens. This weakening is illustrated by a long black candlestick that is unable to close the gap into the body of the first day. These events warn us about a short-term reversal.

EXAMPLE CdlBreakaway(open, high, low, close)

[TOP]

1.3.3.23 CdlClosingmarubozu 0

CDLCLOSINGMARUBOZU
Closing Marubozu

Candlestick Pattern

SYNTAX **CdlClosingmarubozu()**

RETURNS NUMERIC ARRAY

DESCRIPTION Closing Marubozu, a Closing Marubozu has no shadow at it's closing end.

ADDITIONAL INFO
Black Closing marubozu
Signal: Bearish
Pattern: Reversal/Continuation
Reliability: Low
Identification:
The Black Closing Marubozu is a single candlestick pattern characterized by its long black body. It does not have a lower shadow but it has an upper shadow. It is an extremely strong bearish candlestick pattern.
White Closing Marubozu
Signal: Bullish
Pattern: Reversal/Continuation
Reliability: Low
Identification:
The White Closing Marubozu is a single candlestick pattern characterized by a long white body with no upper shadow. This is an extremely strong bullish candlestick pattern.

EXAMPLE CdlClosingmarubozu()

[TOP]

1.3.3.24 CdlClosingmarubozu 1

CDLCLOSINGMARUBOZU
Closing Marubozu

Candlestick Pattern

SYNTAX **CdlClosingmarubozu(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Closing Marubozu, a Closing Marubozu has no shadow at it's closing end.

ADDITIONAL INFO
Black Closing marubozu
Signal: Bearish
Pattern: Reversal/Continuation
Reliability: Low
Identification:
The Black Closing Marubozu is a single candlestick pattern characterized by its long black body. It does not have a lower shadow but it has an upper shadow. It is an extremely strong bearish candlestick pattern.
White Closing Marubozu
Signal: Bullish
Pattern: Reversal/Continuation
Reliability: Low
Identification:
The White Closing Marubozu is a single candlestick pattern characterized by a long white body with no upper shadow. This is an extremely strong bullish candlestick pattern.

EXAMPLE CdlClosingmarubozu(open, high, low, close)

[TOP]

1.3.3.25 CdlConcealbabyswall 0

CDLCONCEALBABYSWALL
Concealing Baby Swallow

Candlestick Pattern

SYNTAX **CdlConcealbabyswall()**

RETURNS NUMERIC ARRAY

DESCRIPTION Concealing Baby Swallow, The first two days of the signal, two Black Marubozus, demonstrate the continuation of the downtrend.

ADDITIONAL INFO
Signal: Bullish
Pattern: reversal
Reliability: high
The first two days are Black Marubozu days (open is the high of the day and the close is the low of the day). The following day is a black day that gaps in the same direction but trades up into the body of the second day. The final day is a Black Marubozu that gaps up and sells off to engulf the third day.

EXAMPLE CdlConcealbabyswall()

[TOP]

1.3.3.26 CdlConcealbabyswall 1

CDLCONCEALBABYSWALL
Concealing Baby Swallow

Candlestick Pattern

SYNTAX	**CdlConcealbabyswall(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Concealing Baby Swallow, The first two days of the signal, two Black Marubozus, demonstrate the continuation of the downtrend.
ADDITIONAL INFO	Signal: Bullish Pattern: reversal Reliability: high The first two days are Black Marubozu days (open is the high of the day and the close is the low of the day). The following day is a black day that gaps in the same direction but trades up into the body of the second day. The final day is a Black Marubozu that gaps up and sells off to engulf the third day.
EXAMPLE	CdlConcealbabyswall(open, high, low, close)

[TOP]

1.3.3.27 CdlCounterattack 0

CDLCOUNTERATTACK
Counterattack Lines

Candlestick Pattern

SYNTAX **CdlCounterattack()**

RETURNS NUMERIC ARRAY

DESCRIPTION Counterattack, Meeting Lines (or Counterattack Lines) are formed when opposite coloured bodies have the same closing price.

ADDITIONAL INFO A counter attack pattern is formed when opposite colour candles share the same clse. This two candle pattern can be bullish or bearish depending where the pattern appears. The counter attack pattern warns that the tide is turning.

EXAMPLE CdlCounterattack()

[TOP]

1.3.3.28 CdlCounterattack 1

CDLCOUNTERATTACK
Counterattack Lines

Candlestick Pattern

SYNTAX **CdlCounterattack(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Counterattack, Meeting Lines (or Counterattack Lines) are formed when opposite coloured bodies have the same closing price.

ADDITIONAL INFO A counter attack pattern is formed when opposite colour candles share the same clse. This two candle pattern can be bullish or bearish depending where the pattern appears. The counter attack pattern warns that the tide is turning.

EXAMPLE CdlCounterattack(open, high, low, close)

[TOP]

1.3.3.29 CdlDarkcloudcover 0

CDLDARKCLOUDCOVER
Dark Cloud Cover

Candlestick Pattern

SYNTAX **CdlDarkcloudcover(ARRAY penetration)**

RETURNS NUMERIC ARRAY

DESCRIPTION Dark Cloud Cover, The dark Cloud Cover is the bearish counterpart to the Piercing pattern.

ADDITIONAL INFO
Dark Cloud Cover Bearish
Signal: Bearish
Pattern: reversal
Reliability: high
Identification
A long white day is followed by a black day which gaps above the high of the white candle and then closes below the midpoint of the first day's body.

EXAMPLE CdlDarkcloudcover(14)

[TOP]

1.3.3.30 CdlDarkcloudcover 1

CDLDARKCLOUDCOVER
Dark Cloud Cover

Candlestick Pattern

SYNTAX	**CdlDarkcloudcover(ARRAY open, ARRAY high, ARRAY low, ARRAY close, ARRAY penetration)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Dark Cloud Cover, The dark Cloud Cover is the bearish counterpart to the Piercing pattern.
ADDITIONAL INFO	Dark Cloud Cover Bearish Signal: Bearish Pattern: reversal Reliability: high Identification A long white day is followed by a black day which gaps above the high of the white candle and then closes below the midpoint of the first day's body.
EXAMPLE	CdlDarkcloudcover(open, high, low, close, 14)

[TOP]

1.3.3.31 CdlDoji 0

CDLDOJI
Doji

Candlestick Pattern

SYNTAX	**CdlDoji()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Doji, the Doji is one of the most important signals in candlestick analysis.
ADDITIONAL INFO	Doji Signal: Indecision Pattern: Reversal/Continuation Reliability: Low Identification: If a security has virtually equal opening and closing prices, this leads to a Doji. The length of the upper and lower shadows of a Doji can vary and consequently the resulting candlestick may look like a cross, inverted cross or a plus sign. Doji, taken alone, is a neutral pattern.
EXAMPLE	CdlDoji()

[TOP]

1.3.3.32 CdlDoji 1

CDLDOJI
Doji

Candlestick Pattern

SYNTAX **CdlDoji(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Doji, the Doji is one of the most important signals in candlestick analysis.

ADDITIONAL INFO
Doji
Signal: Indecision
Pattern: Reversal/Continuation
Reliability: Low
Identification:
If a security has virtually equal opening and closing prices, this leads to a Doji. The length of the upper and lower shadows of a Doji can vary and consequently the resulting candlestick may look like a cross, inverted cross or a plus sign. Doji, taken alone, is a neutral pattern.

EXAMPLE CdlDoji(open, high, low, close)

[TOP]

1.3.3.33 CdlDojistar 0

CDLDOJISTAR
Doji Star

Candlestick Pattern

SYNTAX **CdlDojistar()**

RETURNS NUMERIC ARRAY

DESCRIPTION Doji Star, Upon seeing a Doji in an overbought or oversold condition, an extremely high probability reversal situation becomes evident.

ADDITIONAL INFO
Doji Star Bullish
Signal: Bullish
Pattern: reversal
Reliability: moderate
Identification
A long black day is followed by a Doji that gaps in the direction of the trend. The shadows of the Doji should not be long.
Doji Star Bearish
Signal: Bearish
Pattern: reversal
Reliability: moderate
Identification
A long white day is followed by a Doji that gaps in the direction of the trend. The shadows of the Doji should not be long.

EXAMPLE CdlDojistar()

[TOP]

1.3.3.34 CdlDojistar 1

CDLDOJISTAR
Doji Star

Candlestick Pattern

SYNTAX **CdlDojistar(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Doji Star, Upon seeing a Doji in an overbought or oversold condition, an extremely high probability reversal situation becomes evident.

ADDITIONAL INFO
Doji Star Bullish
Signal: Bullish
Pattern: reversal
Reliability: moderate
Identification
A long black day is followed by a Doji that gaps in the direction of the trend. The shadows of the Doji should not be long.
Doji Star Bearish
Signal: Bearish
Pattern: reversal
Reliability: moderate
Identification
A long white day is followed by a Doji that gaps in the direction of the trend. The shadows of the Doji should not be long.

EXAMPLE CdlDojistar(open, high, low, close)

[TOP]

1.3.3.35 CdlDragonflydoji 0

CDLDRAGONFLYDOJI
Dragonfly Doji

Candlestick Pattern

SYNTAX **CdlDragonflydoji()**

RETURNS NUMERIC ARRAY

DESCRIPTION Dragonfly Doji, The Dragonfly Doji occurs when trading opens, trades lower, then closes at the open price which is the high of the day.

ADDITIONAL INFO Dragonfly Doji Bearish
Signal: Bearish
Pattern: reversal
Reliability: low/moderate
Identification
A Doji forms at the upper end of a trading range with a long lower shadow (the longer the more bearish) with no, or almost no upper shadow.
Dragonfly Doji Bullish
Signal: Bullish
Pattern: reversal
Reliability: low/moderate
Identification
A Doji forms at the upper end of the trading range with a long lower shadow (the longer the more bullish).

EXAMPLE CdlDragonflydoji()

[TOP]

1.3.3.36 CdlDragonflydoji 1

CDLDRAGONFLYDOJI
Dragonfly Doji

Candlestick Pattern

SYNTAX **CdlDragonflydoji(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Dragonfly Doji, The Dragonfly Doji occurs when trading opens, trades lower, then closes at the open price which is the high of the day.

ADDITIONAL INFO
Dragonfly Doji Bearish
Signal: Bearish
Pattern: reversal
Reliability: low/moderate
Identification
A Doji forms at the upper end of a trading range with a long lower shadow (the longer the more bearish) with no, or almost no upper shadow.
Dragonfly Doji Bullish
Signal: Bullish
Pattern: reversal
Reliability: low/moderate
Identification
A Doji forms at the upper end of the trading range with a long lower shadow (the longer the more bullish).

EXAMPLE CdlDragonflydoji(open, high, low, close)

[TOP]

1.3.3.37 CdlEngulfing 0

CDLENGULFING
Engulfing Pattern

Candlestick Pattern

SYNTAX	**CdlEngulfing()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Engulfing Pattern, Two of the most compelling candlestick signals are the Bullish Engulfing Pattern and Bearish Engulfing Pattern.
ADDITIONAL INFO	Engulfing Bullish Signal: Bullish Pattern: reversal Reliability: moderate Identification A black day is completely ï¿½engulfedï¿½ by a large white day that gaps below the black dayï¿½s low and rallies to close above its high. Engulfing Bearish Signal: Bearish Pattern: reversal Reliability: moderate Identification A white day is then completely ï¿½engulfedï¿½ by a large black day which gaps above the white day's high and closes below its low.
EXAMPLE	CdlEngulfing()

[TOP]

1.3.3.38 CdlEngulfing 1

CDLENGULFING
Engulfing Pattern

Candlestick Pattern

SYNTAX **CdlEngulfing(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Engulfing Pattern, Two of the most compelling candlestick signals are the Bullish Engulfing Pattern and Bearish Engulfing Pattern.

ADDITIONAL INFO
Engulfing Bullish
Signal: Bullish
Pattern: reversal
Reliability: moderate
Identification
A black day is completely ï¿½engulfedï¿½ by a large white day that gaps below the black dayï¿½s low and rallies to close above its high.
Engulfing Bearish
Signal: Bearish
Pattern: reversal
Reliability: moderate
Identification
A white day is then completely ï¿½engulfedï¿½ by a large black day which gaps above the white day's high and closes below its low.

EXAMPLE CdlEngulfing(open, high, low, close)

[TOP]

1.3.3.39 CdlEveningdojistar 0

CDLEVENINGDOJISTAR
Evening Doji Star

Candlestick Pattern

SYNTAX	**CdlEveningdojistar(ARRAY penetration)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Evening Doji Star, A three day bearish reversal pattern similar to the Evening Star.
ADDITIONAL INFO	Evening Doji Star Bearish Signal: Bearish Pattern: reversal Reliability: high Identification A long white day is followed by a Doji that gaps in the direction of the trend. The third day is a black day that closes in the bottom half of the white candle.
EXAMPLE	CdlEveningdojistar(14)

[TOP]

1.3.3.40 CdlEveningdojistar 1

CDLEVENINGDOJISTAR
Evening Doji Star

Candlestick Pattern

SYNTAX	**CdlEveningdojistar(ARRAY open, ARRAY high, ARRAY low, ARRAY close, ARRAY penetration)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Evening Doji Star, A three day bearish reversal pattern similar to the Evening Star.
ADDITIONAL INFO	Evening Doji Star Bearish Signal: Bearish Pattern: reversal Reliability: high Identification A long white day is followed by a Doji that gaps in the direction of the trend. The third day is a black day that closes in the bottom half of the white candle.
EXAMPLE	CdlEveningdojistar(open, high, low, close, 14)

[TOP]

1.3.3.41 CdlEveningstar 0

CDLEVENINGSTAR
Evening Star

Candlestick Pattern

SYNTAX **CdlEveningstar(ARRAY penetration)**

RETURNS NUMERIC ARRAY

DESCRIPTION Evening Star, The Evening Star pattern is a top reversal signal.

ADDITIONAL INFO
Evening Star Bearish
Signal: Bearish
Pattern: reversal
Reliability: high
Identification
A long white day is followed by a small body that gaps in the direction of the trend. The third day is a black day that closes in the bottom half of the white candle.

EXAMPLE CdlEveningstar(14)

[TOP]

1.3.3.42 CdlEveningstar 1

CDLEVENINGSTAR
Evening Star

Candlestick Pattern

SYNTAX **CdlEveningstar(ARRAY open, ARRAY high, ARRAY low, ARRAY close, ARRAY penetration)**

RETURNS NUMERIC ARRAY

DESCRIPTION Evening Star, The Evening Star pattern is a top reversal signal.

ADDITIONAL INFO
Evening Star Bearish
Signal: Bearish
Pattern: reversal
Reliability: high
Identification
A long white day is followed by a small body that gaps in the direction of the trend. The third day is a black day that closes in the bottom half of the white candle.

EXAMPLE CdlEveningstar(open, high, low, close, 14)

[TOP]

1.3.3.43 CdlGapsidesidewhite 0

CDLGAPSIDESIDEWHITE
Up/Down-gap side-by-side white lines

Candlestick Pattern

SYNTAX	**CdlGapsidesidewhite()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Up/Down-gap side-by-side white lines, The Up Side By Side White Lines Pattern appears in a bullish market.
ADDITIONAL INFO	Side By Side White Lines Bearish Signal: Bearish Pattern: continuation Reliability: moderate Identification A black day is followed by a white day that gaps in the direction of the trend. The third candle is also white and is almost identical to the previous day. Side by Side White Lines Bullish Signal: Bullish Pattern: reversal Reliability: high Identification A white day is followed by another white day that gaps in the direction of the trend. The third candle is also white and is almost identical to the previous day.
EXAMPLE	CdlGapsidesidewhite()

[TOP]

1.3.3.44 CdlGapsidesidewhite 1

CDLGAPSIDESIDEWHITE
p Side By Side White Lines

Candlestick Pattern

SYNTAX **CdlGapsidesidewhite(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Up/Down-gap side-by-side white lines, The Up Side By Side White Lines Pattern appears in a bullish market.

ADDITIONAL INFO
Side By Side White Lines Bearish
Signal: Bearish
Pattern: continuation
Reliability: moderate
Identification
A black day is followed by a white day that gaps in the direction of the trend. The third candle is also white and is almost identical to the previous day.
Side by Side White Lines Bullish
Signal: Bullish
Pattern: reversal
Reliability: high
Identification
A white day is followed by another white day that gaps in the direction of the trend. The third candle is also white and is almost identical to the previous day.

EXAMPLE CdlGapsidesidewhite(open, high, low, close)

[TOP]

1.3.3.45 CdlGravestonedoji 0

CDLGRAVESTONEDOJI
Gravestone Doji

Candlestick Pattern

SYNTAX **CdlGravestonedoji()**

RETURNS NUMERIC ARRAY

DESCRIPTION Gravestone Doji, The Gravestone Doji is formed when the open and the close occur at the low end of the trading range.

ADDITIONAL INFO
Gravestone Doji Bearish
Signal: Bearish
Pattern: reversal
Reliability: low/moderate
Identification
A Doji forms at the lower end of the trading range. The upper shadow is usually long while the lower shadow is small or almost nonexistent.
Gravestone Doji Bullish
Signal: Bullish
Pattern: reversal
Reliability: low/moderate
Identification
A Doji forms at the lower end of the trading range. The upper shadow is usually long while the lower shadow is small or almost nonexistent.

EXAMPLE CdlGravestonedoji()

[TOP]

1.3.3.46 CdlGravestonedoji 1

CDLGRAVESTONEDOJI
Gravestone Doji

Candlestick Pattern

SYNTAX **CdlGravestonedoji(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Gravestone Doji, The Gravestone Doji is formed when the open and the close occur at the low end of the trading range.

ADDITIONAL INFO
Gravestone Doji Bearish
Signal: Bearish
Pattern: reversal
Reliability: low/moderate
Identification
A Doji forms at the lower end of the trading range. The upper shadow is usually long while the lower shadow is small or almost nonexistent.
Gravestone Doji Bullish
Signal: Bullish
Pattern: reversal
Reliability: low/moderate
Identification
A Doji forms at the lower end of the trading range. The upper shadow is usually long while the lower shadow is small or almost nonexistent.

EXAMPLE CdlGravestonedoji(open, high, low, close)

1.3.3.47 CdlHammer 0

CDLHAMMER
Hammer

Candlestick Pattern

SYNTAX **CdlHammer()**

RETURNS NUMERIC ARRAY

DESCRIPTION Hammer, The Hammer is comprised of one candle.

ADDITIONAL INFO Hammer Bullish
Signal: Bullish
Pattern: reversal
Reliability: low/moderate
Identification
A small real body forms at the upper end of a trading range with a long lower shadow (the longer the more bullish) with no, or almost no upper shadow.

EXAMPLE CdlHammer()

[TOP]

1.3.3.48 CdlHammer 1

CDLHAMMER
Hammer

Candlestick Pattern

SYNTAX	**CdlHammer(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Hammer, The Hammer is comprised of one candle.
ADDITIONAL INFO	Hammer Bullish Signal: Bullish Pattern: reversal Reliability: low/moderate Identification A small real body forms at the upper end of a trading range with a long lower shadow (the longer the more bullish) with no, or almost no upper shadow.
EXAMPLE	CdlHammer(open, high, low, close)

[TOP]

1.3.3.49 CdlHangingman 0

CDLHANGINGMAN
Hanging Man

Candlestick Pattern

SYNTAX	**CdlHangingman()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Hanging Man, The Hanging Man is also comprised of one candle.
ADDITIONAL INFO	Hanging Man Bearish Signal: Bearish Pattern: reversal Reliability: low/moderate Identification A small real body forms at the upper end of the trading range with a long lower shadow (the longer the more bearish) with no, or almost no upper shadow.
EXAMPLE	CdlHangingman()

[TOP]

1.3.3.50 CdlHangingman 1

CDLHANGINGMAN
Hanging Man

Candlestick Pattern

SYNTAX	**CdlHangingman(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Hanging Man, The Hanging Man is also comprised of one candle.
ADDITIONAL INFO	Hanging Man Bearish Signal: Bearish Pattern: reversal Reliability: low/moderate Identification A small real body forms at the upper end of the trading range with a long lower shadow (the longer the more bearish) with no, or almost no upper shadow.
EXAMPLE	CdlHangingman(open, high, low, close)

[TOP]

1.3.3.51 CdlHarami 0

CDLHARAMI
Harami Pattern

Candlestick Pattern

SYNTAX	**CdlHarami()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Harami Pattern, The Harami is an often seen formation.
ADDITIONAL INFO	Harami Bullish Signal: Bullish Pattern: reversal Reliability: low Identification A long black day is followed by a white day which gaps opposite the trend and is completely engulfed by the real body of the first day.
EXAMPLE	CdlHarami()

[TOP]

1.3.3.52 CdlHarami 1

CDLHARAMI
Harami Pattern

Candlestick Pattern

SYNTAX	**CdlHarami(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Harami Pattern, The Harami is an often seen formation.
ADDITIONAL INFO	Harami Bullish Signal: Bullish Pattern: reversal Reliability: low Identification A long black day is followed by a white day which gaps opposite the trend and is completely engulfed by the real body of the first day.
EXAMPLE	CdlHarami(open, high, low, close)

[TOP]

1.3.3.53 CdlHaramicross 0

CDLHARAMICROSS
Harami Cross Pattern

Candlestick Pattern

SYNTAX **CdlHaramicross()**

RETURNS NUMERIC ARRAY

DESCRIPTION Harami Cross Pattern, A two day pattern similar to the Harami (see CDLHARAMI()), the difference is that the last day is a Doji.

ADDITIONAL INFO
Harami Cross Bullish
Signal: Bullish
Pattern: reversal
Reliability: low
Identification
A long black day is followed by a Doji which gaps opposite the trend and is completely engulfed by the real body of the first day.
Harami Cross Bearish
Signal: Bearish
Pattern: reversal
Reliability: moderate
Identification
A long white day is followed by a Doji that gaps down and is completely engulfed by the real body of the first day.

EXAMPLE CdlHaramicross()

[TOP]

1.3.3.54 CdlHaramicross 1

CDLHARAMICROSS
Harami Cross Pattern

Candlestick Pattern

SYNTAX **CdlHaramicross(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Harami Cross Pattern, A two day pattern similar to the Harami (see CDLHARAMI()), the difference is that the last day is a Doji.

ADDITIONAL INFO
Harami Cross Bullish
Signal: Bullish
Pattern: reversal
Reliability: low
Identification
A long black day is followed by a Doji which gaps opposite the trend and is completely engulfed by the real body of the first day.
Harami Cross Bearish
Signal: Bearish
Pattern: reversal
Reliability: moderate
Identification
A long white day is followed by a Doji that gaps down and is completely engulfed by the real body of the first day.

EXAMPLE CdlHaramicross(open, high, low, close)

[TOP]

1.3.3.55 CdlHighwave 0

CDLHIGHWAVE
High-Wave Candle

Candlestick Pattern

SYNTAX **CdlHighwave()**

RETURNS NUMERIC ARRAY

DESCRIPTION High-Wave Candle : This function is contained within the Pattern Recognition set of indicators.

ADDITIONAL INFO
High Wave
Signal: Indecision
Pattern: Reversal
Reliability: Medium
Identification:
High Wave is a type of candlestick characterized with either a very long upper or a lower shadow. It has only a short real body. A group of these patterns may signal a market turn.

EXAMPLE CdlHighwave()

[TOP]

1.3.3.56 CdlHighwave 1

CDLHIGHWAVE
High-Wave Candle

Candlestick Pattern

SYNTAX **CdlHighwave(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION High-Wave Candle : This function is contained within the Pattern Recognition set of indicators.

ADDITIONAL INFO
High Wave
Signal: Indecision
Pattern: Reversal
Reliability: Medium
Identification:
High Wave is a type of candlestick characterized with either a very long upper or a lower shadow. It has only a short real body. A group of these patterns may signal a market turn.

EXAMPLE CdlHighwave(open, high, low, close)

[TOP]

1.3.3.57 CdlHikkake 0

CDLHIKKAKE
Hikkake Pattern

Candlestick Pattern

SYNTAX	**CdlHikkake()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Hikkake Pattern
ADDITIONAL INFO	
EXAMPLE	CdlHikkake()

[TOP]

1.3.3.58 CdlHikkake 1

CDLHIKKAKE
Hikkake Pattern

Candlestick Pattern

SYNTAX	**CdlHikkake(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Hikkake Pattern
ADDITIONAL INFO	
EXAMPLE	CdlHikkake(open, high, low, close)

[TOP]

1.3.3.59 CdlHikkakemod 0

CDLHIKKAKEMOD
Modified Hikkake Pattern

Candlestick Pattern

SYNTAX	**CdlHikkakemod()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Modified Hikkake Pattern
ADDITIONAL INFO	
EXAMPLE	CdlHikkakemod()

[TOP]

1.3.3.60 CdlHikkakemod 1

CDLHIKKAKEMOD
Modified Hikkake Pattern

Candlestick Pattern

SYNTAX	**CdlHikkakemod(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Modified Hikkake Pattern
ADDITIONAL INFO	
EXAMPLE	CdlHikkakemod(open, high, low, close)

[TOP]

1.3.3.61 CdlHomingpigeon 0

CDLHOMINGPIGEON
Homing Pigeon

Candlestick Pattern

SYNTAX	**CdlHomingpigeon()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Homing Pigeon, The Homing Pigeon is the same as the Harami, except for the colour of the second day's body.
ADDITIONAL INFO	Homing Pigeon bullish Signal: Bullish Pattern: reversal Reliability: moderate Identification A long black day is followed by another black day which gaps opposite the trend and is completely engulfed by the real body of the first day.
EXAMPLE	CdlHomingpigeon()

[TOP]

1.3.3.62 CdlHomingpigeon 1

CDLHOMINGPIGEON
Homing Pigeon

Candlestick Pattern

SYNTAX	**CdlHomingpigeon(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Homing Pigeon, The Homing Pigeon is the same as the Harami, except for the colour of the second day's body.
ADDITIONAL INFO	Homing Pigeon bullish Signal: Bullish Pattern: reversal Reliability: moderate Identification A long black day is followed by another black day which gaps opposite the trend and is completely engulfed by the real body of the first day.
EXAMPLE	CdlHomingpigeon(open, high, low, close)

[TOP]

1.3.3.63 CdlIdentical3crows 0

CDLIDENTICAL3CROWS
Identical Three Crows

Candlestick Pattern

SYNTAX	**CdlIdentical3crows()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Identical Three Crows, The Three Identical Crows have the same criteria as the Three Black Crows.
ADDITIONAL INFO	Identical Three Crows Bearish Signal: Bearish Pattern: reversal Reliability: high Identification Three black days with each day opening where the previous day closed.
EXAMPLE	CdlIdentical3crows()

[TOP]

1.3.3.64 CdlIdentical3crows 1

CDLIDENTICAL3CROWS
Identical Three Crows

Candlestick Pattern

SYNTAX **CdlIdentical3crows(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Identical Three Crows, The Three Identical Crows have the same criteria as the Three Black Crows.

ADDITIONAL INFO Identical Three Crows Bearish
Signal: Bearish
Pattern: reversal
Reliability: high
Identification
Three black days with each day opening where the previous day closed.

EXAMPLE CdlIdentical3crows(open, high, low, close)

[TOP]

1.3.3.65 CdlInneck 0

CDLINNECK
In-Neck Pattern

Candlestick Pattern

SYNTAX	**CdlInneck()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	In-Neck Pattern, The In Neck pattern is almost a Meeting Line pattern.
ADDITIONAL INFO	In Neck Bearish Signal: Bearish Pattern: continuation Reliability: moderate Identification A long black day is followed by a long white day that gaps down at the open and closes at the same price as the black day.
EXAMPLE	CdlInneck()

[TOP]

1.3.3.66 CdlInneck 1

CDLINNECK
In-Neck Pattern

Candlestick Pattern

SYNTAX **CdlInneck(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION In-Neck Pattern, The In Neck pattern is almost a Meeting Line pattern.

ADDITIONAL INFO In Neck Bearish
Signal: Bearish
Pattern: continuation
Reliability: moderate
Identification
A long black day is followed by a long white day that gaps down at the open and closes at the same price as the black day.

EXAMPLE CdlInneck(open, high, low, close)

[TOP]

1.3.3.67 CdlInvertedhammer 0

CDLINVERTEDHAMMER
Inverted Hammer

Candlestick Pattern

SYNTAX **CdlInvertedhammer()**

RETURNS NUMERIC ARRAY

DESCRIPTION Inverted Hammer, The Inverted Hammer is comprised of one candle.

ADDITIONAL INFO Inverted Hammer Bullish
Signal: Bullish
Pattern: reversal
Reliability: low/moderate
Identification
A small real body forms at the lower end of the trading range. The upper shadow is usually no more than twice as long as the real body and there is no or almost no lower shadow.

EXAMPLE CdlInvertedhammer()

[TOP]

1.3.3.68 CdlInvertedhammer 1

CDLINVERTEDHAMMER
Inverted Hammer

Candlestick Pattern

SYNTAX **CdlInvertedhammer(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Inverted Hammer, The Inverted Hammer is comprised of one candle.

ADDITIONAL INFO Inverted Hammer Bullish
Signal: Bullish
Pattern: reversal
Reliability: low/moderate
Identification
A small real body forms at the lower end of the trading range. The upper shadow is usually no more than twice as long as the real body and there is no or almost no lower shadow.

EXAMPLE CdlInvertedhammer(open, high, low, close)

[TOP]

1.3.3.69 CdlKicking 0

CDLKICKING
Kicking

Candlestick Pattern

SYNTAX **CdlKicking()**

RETURNS NUMERIC ARRAY

DESCRIPTION Kicking, The Kicker signal is the most powerful signal of all.

ADDITIONAL INFO
Kicking Bearish
Signal: Bearish
Pattern: reversal
Reliability: high
Identification
A White Marubuzo (opens at low and closes at high) is followed by a Black Marubuzo (opens at high and closes at low) that gaps down.
Kicking Bullish
Signal: Bullish
Pattern: reversal
Reliability: high
Identification:
A Black Marubuzo (open is the high of the day and the close is the low of the day) day is followed by a White Marubuzo (open is the low of the day and the close is the high of the day) day that gaps in the opposite direction.

EXAMPLE CdlKicking()

[TOP]

1.3.3.70 CdlKicking 1

CDLKICKING
Kicking

Candlestick Pattern

SYNTAX **CdlKicking(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Kicking, The Kicker signal is the most powerful signal of all.

ADDITIONAL INFO
Kicking Bearish
Signal: Bearish
Pattern: reversal
Reliability: high
Identification
A White Marubuzo (opens at low and closes at high) is followed by a Black Marubuzo (opens at high and closes at low) that gaps down.
Kicking Bullish
Signal: Bullish
Pattern: reversal
Reliability: high
Identification:
A Black Marubuzo (open is the high of the day and the close is the low of the day) day is followed by a White Marubuzo (open is the low of the day and the close is the high of the day) day that gaps in the opposite direction.

EXAMPLE CdlKicking(open, high, low, close)

[TOP]

1.3.3.71 CdlKickingbylength 0

CDLKICKINGBYLENGTH
Kicking - bull/bear

Candlestick Pattern

SYNTAX **CdlKickingbylength()**

RETURNS NUMERIC ARRAY

DESCRIPTION Kicking - bull/bear determined by the longer marubozu, The Kicker signal is the most powerful signal of all.

ADDITIONAL INFO

EXAMPLE CdlKickingbylength()

[TOP]

1.3.3.72 CdlKickingbylength 1

CDLKICKINGBYLENGTH
Kicking - bull/bear

Candlestick Pattern

SYNTAX	**CdlKickingbylength(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Kicking - bull/bear determined by the longer marubozu, The Kicker signal is the most powerful signal of all.
ADDITIONAL INFO	
EXAMPLE	CdlKickingbylength(open, high, low, close)

[TOP]

1.3.3.73 CdlLadderbottom 0

CDLLADDERBOTTOM
Ladder Bottom

Candlestick Pattern

SYNTAX	**CdlLadderbottom()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Ladder Bottom, The downtrend is finishing with four consecutive black candles, each closing lower than the previous day.
ADDITIONAL INFO	Ladder Bottom Bullish Signal: Bullish Pattern: reversal Reliability: moderate Identification Three black days with successive lower opens and lower closes. Then a black day forms with some noticeably upper shadow. The final day is a white day that gaps against the trend and opens above the body of the fourth day.
EXAMPLE	CdlLadderbottom()

[TOP]

1.3.3.74 CdlLadderbottom 1

CDLLADDERBOTTOM
Ladder Bottom

Candlestick Pattern

SYNTAX **CdlLadderbottom(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Ladder Bottom, The downtrend is finishing with four consecutive black candles, each closing lower than the previous day.

ADDITIONAL INFO
Ladder Bottom Bullish
Signal: Bullish
Pattern: reversal
Reliability: moderate
Identification
Three black days with successive lower opens and lower closes. Then a black day forms with some noticeably upper shadow. The final day is a white day that gaps against the trend and opens above the body of the fourth day.

EXAMPLE CdlLadderbottom(open, high, low, close)

[TOP]

1.3.3.75 CdlLongleggeddoji 0

CDLLONGLEGGEDDOJI
Long Legged Doji

Candlestick Pattern

SYNTAX **CdlLongleggeddoji()**

RETURNS NUMERIC ARRAY

DESCRIPTION Long Legged Doji, The Long-legged Doji is composed of long upper and lower shadows.

ADDITIONAL INFO
Long Legged Doji
Signal: Indecision
Pattern: Reversal
Reliability: Medium
Identification:
Long Legged Doji is characterized by very long shadows. It is an important reversal signal.
Bullish Long Legged Doji
Signal: Bullish
Pattern: Reversal
Reliability: Medium
Identification:
Long Legged Doji is a doji characterized with very long shadows. It shows the indecision of the buyers and sellers. It is one of the important reversal signals.
Bearish Long Legged Doji
Signal: Bearish
Pattern: Reversal
Reliability: Medium
Identification:
Long Legged Doji is a doji characterized by very long shadows. It shows the indecision of the buyers and sellers and it is an important reversal signal.

EXAMPLE CdlLongleggeddoji()

[TOP]

1.3.3.76 CdlLongleggeddoji 1

CDLLONGLEGGEDDOJI
Long Legged Doji

Candlestick Pattern

SYNTAX	**CdlLongleggeddoji(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Long Legged Doji, The Long-legged Doji is composed of long upper and lower shadows.
ADDITIONAL INFO	Long Legged Doji Signal: Indecision Pattern: Reversal Reliability: Medium Identification: Long Legged Doji is characterized by very long shadows. It is an important reversal signal. Bullish Long Legged Doji Signal: Bullish Pattern: Reversal Reliability: Medium Identification: Long Legged Doji is a doji characterized with very long shadows. It shows the indecision of the buyers and sellers. It is one of the important reversal signals. Bearish Long Legged Doji Signal: Bearish Pattern: Reversal Reliability: Medium Identification: Long Legged Doji is a doji characterized by very long shadows. It shows the indecision of the buyers and sellers and it is an important reversal signal.
EXAMPLE	CdlLongleggeddoji(open, high, low, close)

[TOP]

1.3.3.77 CdlLongline 0

CDLLONGLINE
Long Line Candle

Candlestick Pattern

SYNTAX	**CdlLongline()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Long Line Candle, A long day (LONGLINE) represents a large price move from open to close.
ADDITIONAL INFO	Long Line Candle Signal: Indecision Pattern: Reversal Reliability: Low Identification: A long day (Long Line Candle) represents a large price move from open to close.
EXAMPLE	CdlLongline()

[TOP]

1.3.3.78 CdlLongline 1

CDLLONGLINE
Long Line Candle

Candlestick Pattern

SYNTAX **CdlLongline(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Long Line Candle, A long day (LONGLINE) represents a large price move from open to close.

ADDITIONAL INFO
Long Line Candle
Signal: Indecision
Pattern: Reversal
Reliability: Low
Identification:
A long day (Long Line Candle) represents a large price move from open to close.

EXAMPLE CdlLongline(open, high, low, close)

[TOP]

1.3.3.79 CdlMarubozu 0

CDLMARUBOZU
Marubozu

Candlestick Pattern

SYNTAX **CdlMarubozu()**

RETURNS NUMERIC ARRAY

DESCRIPTION Marubozu, in Japanese, Marubozu means close cropped or close-cut.

ADDITIONAL INFO
Black Marubozu
Signal: Bearish
Pattern: Reversal/Continuation
Reliability: Low
Identification:
The Black Marubozu is a single candlestick pattern characterized by a long black body. It does not have any shadows on either end. It is an extremely strong bearish candlestick pattern.
White Marubozu
Signal: Bullish
Pattern: Reversal/Continuation
Reliability: Low
Identification:
The White Marubozu is a single candlestick pattern characterized with a long white body having no shadows on either end. It is an extremely strong bullish candlestick pattern.

EXAMPLE CdlMarubozu()

[TOP]

1.3.3.80 CdlMarubozu 1

CDLMARUBOZU
Marubozu

Candlestick Pattern

SYNTAX **CdlMarubozu(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Marubozu, in Japanese, Marubozu means close cropped or close-cut.

ADDITIONAL INFO
Black Marubozu
Signal: Bearish
Pattern: Reversal/Continuation
Reliability: Low
Identification:
The Black Marubozu is a single candlestick pattern characterized by a long black body. It does not have any shadows on either end. It is an extremely strong bearish candlestick pattern.
White Marubozu
Signal: Bullish
Pattern: Reversal/Continuation
Reliability: Low
Identification:
The White Marubozu is a single candlestick pattern characterized with a long white body having no shadows on either end. It is an extremely strong bullish candlestick pattern.

EXAMPLE CdlMarubozu(open, high, low, close)

[TOP]

1.3.3.81 CdlMatchinglow 0

CDLMATCHINGLOW
Matching Low

Candlestick Pattern

SYNTAX **CdlMatchinglow()**

RETURNS NUMERIC ARRAY

DESCRIPTION Matching Low, The Matching Low pattern is similar to the Homing Pigeon patter, the exception being that the two days of the pattern close on their lows, at the same level.

ADDITIONAL INFO
Matching Low Bullish
Signal: Bullish
Pattern: reversal
Reliability: high
Identification
A long black day is followed by another black day with equivalent closes both days.

EXAMPLE CdlMatchinglow()

[TOP]

1.3.3.82 CdlMatchinglow 1

CDLMATCHINGLOW
Matching Low

Candlestick Pattern

SYNTAX **CdlMatchinglow(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Matching Low, The Matching Low pattern is similar to the Homing Pigeon patter, the exception being that the two days of the pattern close on their lows, at the same level.

ADDITIONAL INFO Matching Low Bullish
Signal: Bullish
Pattern: reversal
Reliability: high
Identification
A long black day is followed by another black day with equivalent closes both days.

EXAMPLE CdlMatchinglow(open, high, low, close)

[TOP]

1.3.3.83 CdlMathold 0

CDLMATHOLD
Mat Hold

Candlestick Pattern

SYNTAX	**CdlMathold(ARRAY penetration)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Mat Hold, The pattern appears during an uptrend, which is further confirmed by the first long white candlestick.
ADDITIONAL INFO	Mat Hold Bullish Signal: Bullish Pattern: continuation Reliability: high Identification A long white day in an uptrend is followed by a relatively small black day that gaps in the direction of the trend. The next two days continue the brief pullback and are small days that stay within the range of the first day. The fifth day is a long white day that closes above the close of the first day and continues the uptrend.
EXAMPLE	CdlMathold(14)

[TOP]

1.3.3.84 CdlMathold 1

CDLMATHOLD
Mat Hold

Candlestick Pattern

SYNTAX **CdlMathold(ARRAY open, ARRAY high, ARRAY low, ARRAY close, ARRAY penetration)**

RETURNS NUMERIC ARRAY

DESCRIPTION Mat Hold, The pattern appears during an uptrend, which is further confirmed by the first long white candlestick.

ADDITIONAL INFO
Mat Hold Bullish
Signal: Bullish
Pattern: continuation
Reliability: high
Identification
A long white day in an uptrend is followed by a relatively small black day that gaps in the direction of the trend. The next two days continue the brief pullback and are small days that stay within the range of the first day. The fifth day is a long white day that closes above the close of the first day and continues the uptrend.

EXAMPLE CdlMathold(open, high, low, close, 14)

[TOP]

1.3.3.85 CdlMorningdojistar 0

CDLMORNINGDOJISTAR
Morning Doji Star

Candlestick Pattern

SYNTAX **CdlMorningdojistar(ARRAY penetration)**

RETURNS NUMERIC ARRAY

DESCRIPTION Morning Doji Star, A three day bullish reversal pattern that is very similar to the Morning Star.

ADDITIONAL INFO Morning Doji Star Bullish
Signal: Bullish
Pattern: reversal
Reliability: high
Identification
A long black day is followed by a Doji that gaps in the direction of the trend. The third day is a white day which closes in the top half of the black day.

EXAMPLE CdlMorningdojistar(14)

[TOP]

1.3.3.86 CdlMorningdojistar 1

CDLMORNINGDOJISTAR
Morning Doji Star

Candlestick Pattern

SYNTAX **CdlMorningdojistar(ARRAY open, ARRAY high, ARRAY low, ARRAY close, ARRAY penetration)**

RETURNS NUMERIC ARRAY

DESCRIPTION Morning Doji Star, A three day bullish reversal pattern that is very similar to the Morning Star.

ADDITIONAL INFO
Morning Doji Star Bullish
Signal: Bullish
Pattern: reversal
Reliability: high
Identification
A long black day is followed by a Doji that gaps in the direction of the trend. The third day is a white day which closes in the top half of the black day.

EXAMPLE CdlMorningdojistar(open, high, low, close, 14)

[TOP]

1.3.3.87 CdlMorningstar 0

CDLMORNINGSTAR
Morning Star

Candlestick Pattern

SYNTAX **CdlMorningstar(ARRAY penetration)**

RETURNS NUMERIC ARRAY

DESCRIPTION Morning Star, The Morning Star is a bottom reversal signal.

ADDITIONAL INFO
Morning Star Bullish
Signal: Bullish
Pattern: reversal
Reliability: high
Identification
A long black day is followed by a small day that gaps in the direction of the trend. The third day is a white day which closes in the top half of the black day.

EXAMPLE CdlMorningstar(14)

[TOP]

1.3.3.88 CdlMorningstar 1

CDLMORNINGSTAR
Morning Star

Candlestick Pattern

SYNTAX **CdlMorningstar(ARRAY open, ARRAY high, ARRAY low, ARRAY close, ARRAY penetration)**

RETURNS NUMERIC ARRAY

DESCRIPTION Morning Star, The Morning Star is a bottom reversal signal.

ADDITIONAL INFO
Morning Star Bullish
Signal: Bullish
Pattern: reversal
Reliability: high
Identification
A long black day is followed by a small day that gaps in the direction of the trend. The third day is a white day which closes in the top half of the black day.

EXAMPLE CdlMorningstar(open, high, low, close, 14)

[TOP]

1.3.3.89 CdlOnneck 0

CDLONNECK
On-Neck Pattern

Candlestick Pattern

SYNTAX **CdlOnneck()**

RETURNS NUMERIC ARRAY

DESCRIPTION On-Neck Pattern, The On Neck Line pattern is almost a 'meeting line pattern', but the critical term is 'almost'.

ADDITIONAL INFO On Neck Bearish
Signal: Bearish
Pattern: continuation
Reliability: moderate
Identification
A long black day is followed by a long white day that gaps down at the open and closes below the close of the black day.

EXAMPLE CdlOnneck()

[TOP]

1.3.3.90 CdlOnneck 1

CDLONNECK
On-Neck Pattern

Candlestick Pattern

SYNTAX	**CdlOnneck(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	On-Neck Pattern, The On Neck Line pattern is almost a 'meeting line pattern', but the critical term is 'almost'.
ADDITIONAL INFO	On Neck Bearish Signal: Bearish Pattern: continuation Reliability: moderate Identification A long black day is followed by a long white day that gaps down at the open and closes below the close of the black day.
EXAMPLE	CdlOnneck(open, high, low, close)

[TOP]

1.3.3.91 CdlPiercing 0

CDLPIERCING
Piercing Pattern

Candlestick Pattern

SYNTAX	**CdlPiercing()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Piercing Pattern, The Piercing Pattern is composed of a two-candle formation in a down trending market.
ADDITIONAL INFO	Piercing Line Bullish Signal: Bullish Pattern: reversal Reliability: moderate Identification A long black day is followed by a white day that gaps below the black dayï¿½s low and closes within and above the midpoint of the black dayï¿½s body.
EXAMPLE	CdlPiercing()

[TOP]

1.3.3.92 CdlPiercing 1

CDLPIERCING
Piercing Pattern

Candlestick Pattern

SYNTAX **CdlPiercing(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Piercing Pattern, The Piercing Pattern is composed of a two-candle formation in a down trending market.

ADDITIONAL INFO Piercing Line Bullish
Signal: Bullish
Pattern: reversal
Reliability: moderate
Identification
A long black day is followed by a white day that gaps below the black dayï¿½s low and closes within and above the midpoint of the black dayï¿½s body.

EXAMPLE CdlPiercing(open, high, low, close)

1.3.3.93 CdlRickshawman 0

CDLRICKSHAWMAN
Rickshaw Man

Candlestick Pattern

SYNTAX	**CdlRickshawman()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Rickshaw Man, Rickshaw man is a specific type of long-legged doji where the open and close are in the middle of the price range.
ADDITIONAL INFO	Rickshaw Man Signal: Indecision Pattern: Reversal Reliability: Medium Identification: Rickshaw man is a specific type of long-legged doji where the open and close are in the middle of the price range.
EXAMPLE	CdlRickshawman()

[TOP]

1.3.3.94 CdlRickshawman 1

CDLRICKSHAWMAN
Rickshaw Man

Candlestick Pattern

SYNTAX	**CdlRickshawman(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Rickshaw Man, Rickshaw man is a specific type of long-legged doji where the open and close are in the middle of the price range.
ADDITIONAL INFO	Rickshaw Man Signal: Indecision Pattern: Reversal Reliability: Medium Identification: Rickshaw man is a specific type of long-legged doji where the open and close are in the middle of the price range.
EXAMPLE	CdlRickshawman(open, high, low, close)

[TOP]

1.3.3.95 CdlRisefall3methods 0

CDLRISEFALL3METHODS
Rising/Falling Three Methods

Candlestick Pattern

SYNTAX **CdlRisefall3methods()**

RETURNS NUMERIC ARRAY

DESCRIPTION Rising/Falling Three Methods, The Falling Three Method is basically the opposite of the Rising Three Method, The market has been in a downtrend.

ADDITIONAL INFO
Rising Three Methods Bullish
Signal: Bullish
Pattern: continuation
Reliability: high
Identification
A long white day in an uptrend is followed by three relatively small candles that move opposite the overall trend but stay within the range of the first day. The fifth day is a long white day that closes above the close of the first day and continues the uptrend.
Fallling Three Methods Bearish
Signal: Bearish
Pattern: continuation
Reliability: high
Identification
A long black day in a downtrend is followed by three relatively small candles that move opposite the overall trend but stay within the range of the first day. The fifth day is a long black day that closes below the close of the first day and continues the downtrend.

EXAMPLE CdlRisefall3methods()

[TOP]

1.3.3.96 CdlRisefall3methods 1

CDLRISEFALL3METHODS
Rising/Falling Three Methods

Candlestick Pattern

SYNTAX **CdlRisefall3methods(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Rising/Falling Three Methods, The Falling Three Method is basically the opposite of the Rising Three Method, The market has been in a downtrend.

ADDITIONAL INFO
Rising Three Methods Bullish
Signal: Bullish
Pattern: continuation
Reliability: high
Identification
A long white day in an uptrend is followed by three relatively small candles that move opposite the overall trend but stay within the range of the first day. The fifth day is a long white day that closes above the close of the first day and continues the uptrend.
Fallling Three Methods Bearish
Signal: Bearish
Pattern: continuation
Reliability: high
Identification
A long black day in a downtrend is followed by three relatively small candles that move opposite the overall trend but stay within the range of the first day. The fifth day is a long black day that closes below the close of the first day and continues the downtrend.

EXAMPLE CdlRisefall3methods(open, high, low, close)

[TOP]

1.3.3.97 CdlSeparatinglines 0

CDLSEPARATINGLINES
Separating Lines

Candlestick Pattern

SYNTAX **CdlSeparatinglines()**

RETURNS NUMERIC ARRAY

DESCRIPTION Separating Lines, You can identify it from the following points: The first day is a long white candle.

ADDITIONAL INFO
Separating Lines Bullish
Signal: Bullish
Pattern: continuation
Reliability: low
Identification
A black day is followed by a white day that has the same opening price.
Separating Lines Bearish
Signal: Bearish
Pattern: continuation
Reliability: low
Identification
A white day is followed by a black day that has the same opening price.

EXAMPLE CdlSeparatinglines()

[TOP]

1.3.3.98 CdlSeparatinglines 1

CDLSEPARATINGLINES
Separating Lines

Candlestick Pattern

SYNTAX **CdlSeparatinglines(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Separating Lines, You can identify it from the following points: The first day is a long white candle.

ADDITIONAL INFO
Separating Lines Bullish
Signal: Bullish
Pattern: continuation
Reliability: low
Identification
A black day is followed by a white day that has the same opening price.
Separating Lines Bearish
Signal: Bearish
Pattern: continuation
Reliability: low
Identification
A white day is followed by a black day that has the same opening price.

EXAMPLE CdlSeparatinglines(open, high, low, close)

[TOP]

1.3.3.99 CdlShootingstar 0

CDLSHOOTINGSTAR
Shooting Star

Candlestick Pattern

SYNTAX **CdlShootingstar()**

RETURNS NUMERIC ARRAY

DESCRIPTION Shooting Star, The Shooting Star is comprised of one candle.

ADDITIONAL INFO
Shooting Star Bearish
Signal: Bearish
Pattern: reversal
Reliability: low/moderate
Identification
A small body forms at the lower end of the trading range. The upper shadow is usually long while the lower shadow is small or almost nonexistent.

EXAMPLE CdlShootingstar()

[TOP]

1.3.3.100 CdlShootingstar 1

CDLSHOOTINGSTAR
Shooting Star

Candlestick Pattern

SYNTAX **CdlShootingstar(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Shooting Star, The Shooting Star is comprised of one candle.

ADDITIONAL INFO
Shooting Star Bearish
Signal: Bearish
Pattern: reversal
Reliability: low/moderate
Identification
A small body forms at the lower end of the trading range. The upper shadow is usually long while the lower shadow is small or almost nonexistent.

EXAMPLE CdlShootingstar(open, high, low, close)

[TOP]

1.3.3.101 CdlShortline 0

CDLSHORTLINE
Short Line Candle

Candlestick Pattern

SYNTAX **CdlShortline()**

RETURNS NUMERIC ARRAY

DESCRIPTION Short Line Candle, Short days (SHORTLINES) can be interpreted by the same analytical process of the long candles.

ADDITIONAL INFO Short Line Candle
Signal: Indecision
Pattern: Reversal
Reliability: Low
Identification:
Short days (Short Line Candle) represents a small price move from open to close.

EXAMPLE CdlShortline()

[TOP]

1.3.3.102 CdlShortline 1

CDLSHORTLINE
Short Line Candle

Candlestick Pattern

SYNTAX **CdlShortline(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Short Line Candle, Short days (SHORTLINES) can be interpreted by the same analytical process of the long candles.

ADDITIONAL INFO
Short Line Candle
Signal: Indecision
Pattern: Reversal
Reliability: Low
Identification:
Short days (Short Line Candle) represents a small price move from open to close.

EXAMPLE CdlShortline(open, high, low, close)

[TOP]

1.3.3.103 CdlSpinningtop 0

CDLSPINNINGTOP
Spinning Top

Candlestick Pattern

SYNTAX	**CdlSpinningtop()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Spinning Top, spinning Tops are depicted with small bodies relative to the shadows.
ADDITIONAL INFO	White Spinning Top Signal: Indecision Pattern: Reversal/Continuation Reliability: Low Identification: The White Spinning Top is a single candlestick pattern. Its shape is a small white body with upper and lower shadows that have a greater length than the body's length. Black Spinning Top Signal: Indecision Pattern: Reversal/Continuation Reliability: Low Identification: The Black Spinning Top is a single candlestick pattern. Its shape is a small black body with upper and lower shadows having a greater length than the body's length.
EXAMPLE	CdlSpinningtop()

[TOP]

1.3.3.104 CdlSpinningtop 1

CDLSPINNINGTOP
Spinning Top

Candlestick Pattern

SYNTAX	**CdlSpinningtop(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Spinning Top, spinning Tops are depicted with small bodies relative to the shadows.
ADDITIONAL INFO	White Spinning Top Signal: Indecision Pattern: Reversal/Continuation Reliability: Low Identification: The White Spinning Top is a single candlestick pattern. Its shape is a small white body with upper and lower shadows that have a greater length than the body's length. Black Spinning Top Signal: Indecision Pattern: Reversal/Continuation Reliability: Low Identification: The Black Spinning Top is a single candlestick pattern. Its shape is a small black body with upper and lower shadows having a greater length than the body's length.
EXAMPLE	CdlSpinningtop(open, high, low, close)

[TOP]

1.3.3.105 CdlStalledpattern 0

CDLSTALLEDPATTERN
Stalled Pattern

Candlestick Pattern

SYNTAX	**CdlStalledpattern()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Stalled Pattern, Another pattern close to the Three White Soldiers pattern is the Stalled Pattern (commonly known as the Deliberation pattern).
ADDITIONAL INFO	Stalled Pattern or Deliberation pattern Signal: Bearish Pattern: Reversal Reliability: Medium Identification: The Bearish Deliberation Pattern is a derivative of the Bearish Three White Soldiers Pattern. This pattern also shows a weakness similar to the Bearish Advance Block Pattern since it becomes weaker in a short period of time. However here the weakness occurs all at once on the third day. The small third body of the pattern shows that the rally is losing strength and a reversal is possible.
EXAMPLE	CdlStalledpattern()

[TOP]

1.3.3.106 CdlStalledpattern 1

CDLSTALLEDPATTERN
Stalled Pattern

Candlestick Pattern

SYNTAX **CdlStalledpattern(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Stalled Pattern, Another pattern close to the Three White Soldiers pattern is the Stalled Pattern (commonly known as the Deliberation pattern).

ADDITIONAL INFO Stalled Pattern or Deliberation pattern
Signal: Bearish
Pattern: Reversal
Reliability: Medium
Identification:
The Bearish Deliberation Pattern is a derivative of the Bearish Three White Soldiers Pattern. This pattern also shows a weakness similar to the Bearish Advance Block Pattern since it becomes weaker in a short period of time. However here the weakness occurs all at once on the third day. The small third body of the pattern shows that the rally is losing strength and a reversal is possible.

EXAMPLE CdlStalledpattern(open, high, low, close)

[TOP]

1.3.3.107 CdlSticksandwich 0

CDLSTICKSANDWICH
Stick Sandwich

Candlestick Pattern

SYNTAX **CdlSticksandwich()**

RETURNS NUMERIC ARRAY

DESCRIPTION Stick Sandwich, The Stick Sandwich looks somewhat like an ice cream sandwich.

ADDITIONAL INFO
Bullish Stick Sandwich
Signal: Bullish
Pattern: Reversal
Reliability: Medium
Identification:
The Bullish Stick Sandwich Pattern is characterized by consecutive higher opens for three days, but results in an eventual close equal to the first day's close. It may warn that prices are now finding a support price. We may then see a reversal from this support level.

EXAMPLE CdlSticksandwich()

[TOP]

1.3.3.108 CdlSticksandwich 1

CDLSTICKSANDWICH
Stick Sandwich

Candlestick Pattern

SYNTAX	**CdlSticksandwich(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Stick Sandwich, The Stick Sandwich looks somewhat like an ice cream sandwich.
ADDITIONAL INFO	Bullish Stick Sandwich Signal: Bullish Pattern: Reversal Reliability: Medium Identification: The Bullish Stick Sandwich Pattern is characterized by consecutive higher opens for three days, but results in an eventual close equal to the first day's close. It may warn that prices are now finding a support price. We may then see a reversal from this support level.
EXAMPLE	CdlSticksandwich(open, high, low, close)

[TOP]

1.3.3.109 CdlTakuri 0

CDLTAKURI
Takuri

Candlestick Pattern

SYNTAX	**CdlTakuri()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Takuri (Dragonfly Doji with very long lower shadow) : This function is contained within the Pattern Recognition set of indicators.
ADDITIONAL INFO	Dragonfly Doji with very long lower shadow
EXAMPLE	CdlTakuri()

[TOP]

1.3.3.110 CdlTakuri 1

CDLTAKURI
Takuri

Candlestick Pattern

SYNTAX	**CdlTakuri(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Takuri (Dragonfly Doji with very long lower shadow) : This function is contained within the Pattern Recognition set of indicators.
ADDITIONAL INFO	Dragonfly Doji with very long lower shadow
EXAMPLE	CdlTakuri(open, high, low, close)

[TOP]

1.3.3.111 CdlTasukigap 0

CDLTASUKIGAP
Tasuki Gap

Candlestick Pattern

SYNTAX **CdlTasukigap()**

RETURNS NUMERIC ARRAY

DESCRIPTION Tasuki Gap, The Upside Tasuki Gap is found in a rising trend.

ADDITIONAL INFO
Downside Tasuki Gap Bearish
Signal: Bearish
Pattern: continuation
Reliability: moderate
Identification
A long black day is followed by a second long black day that gaps in the direction of the trend. The third day is white and opens within the body of the second day and closes within the gap.
Upside Tasuki Gap Bullish
Signal: Bullish
Pattern: continuation
Reliability: moderate
Identification
A long white day is followed by a second long white day that gaps in the direction of the trend. The third day is black and opens within the body of the second day and closes within the gap.

EXAMPLE CdlTasukigap()

[TOP]

1.3.3.112 CdlTasukigap 1

CDLTASUKIGAP
Tasuki Gap

Candlestick Pattern

SYNTAX **CdlTasukigap(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Tasuki Gap, The Upside Tasuki Gap is found in a rising trend.

ADDITIONAL INFO
Downside Tasuki Gap Bearish
Signal: Bearish
Pattern: continuation
Reliability: moderate
Identification
A long black day is followed by a second long black day that gaps in the direction of the trend. The third day is white and opens within the body of the second day and closes within the gap.
Upside Tasuki Gap Bullish
Signal: Bullish
Pattern: continuation
Reliability: moderate
Identification
A long white day is followed by a second long white day that gaps in the direction of the trend. The third day is black and opens within the body of the second day and closes within the gap.

EXAMPLE CdlTasukigap(open, high, low, close)

[TOP]

1.3.3.113 CdlThrusting 0

CDLTHRUSTING
Thrusting Pattern

Candlestick Pattern

SYNTAX	**CdlThrusting()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Thrusting Pattern, The Thrusting pattern is almost an 'On Neck' or an 'In Neck' pattern and resembles the Meeting Line pattern, also.
ADDITIONAL INFO	Thrusting Bearish Signal: Bearish Pattern: continuation Reliability: low Identification A black day is followed by a white day which gaps in the direction of the trend and closes below the midpoint of the black day.
EXAMPLE	CdlThrusting()

[TOP]

1.3.3.114 CdlThrusting 1

CDLTHRUSTING
Thrusting Pattern

Candlestick Pattern

SYNTAX **CdlThrusting(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Thrusting Pattern, The Thrusting pattern is almost an 'On Neck' or an 'In Neck' pattern and resembles the Meeting Line pattern, also.

ADDITIONAL INFO
Thrusting Bearish
Signal: Bearish
Pattern: continuation
Reliability: low
Identification
A black day is followed by a white day which gaps in the direction of the trend and closes below the midpoint of the black day.

EXAMPLE CdlThrusting(open, high, low, close)

[TOP]

1.3.3.115 CdlTristar 0

CDLTRISTAR
Tristar Pattern

Candlestick Pattern

SYNTAX	**CdlTristar()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Tristar Pattern, The Tri Star pattern is relatively rare.
ADDITIONAL INFO	
EXAMPLE	CdlTristar()

[TOP]

1.3.3.116 CdlTristar 1

CDLTRISTAR
Tristar Pattern

Candlestick Pattern

SYNTAX	**CdlTristar(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Tristar Pattern, The Tri Star pattern is relatively rare.
ADDITIONAL INFO	
EXAMPLE	CdlTristar(open, high, low, close)

[TOP]

1.3.3.117 CdlUnique3river 0

CDLUNIQUE3RIVER
Unique 3 River

Candlestick Pattern

SYNTAX **CdlUnique3river()**

RETURNS NUMERIC ARRAY

DESCRIPTION Unique 3 River, The Unique Three River Bottom is a bullish pattern, somewhat characteristic of the Morning Star Pattern.

ADDITIONAL INFO
Unique Three River Bottom Bullish
Signal: Bullish
Pattern: reversal
Reliability: moderate
Identification
The first day is a long black day, followed by a Homing Pigeon whose lower shadow makes a new low. The last day is small and white, and closes below the second dayï¿½s close.

EXAMPLE CdlUnique3river()

[TOP]

1.3.3.118 CdlUnique3river 1

CDLUNIQUE3RIVER
Unique 3 River

Candlestick Pattern

SYNTAX **CdlUnique3river(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Unique 3 River, The Unique Three River Bottom is a bullish pattern, somewhat characteristic of the Morning Star Pattern.

ADDITIONAL INFO
Unique Three River Bottom Bullish
Signal: Bullish
Pattern: reversal
Reliability: moderate
Identification
The first day is a long black day, followed by a Homing Pigeon whose lower shadow makes a new low. The last day is small and white, and closes below the second dayï¿½s close.

EXAMPLE CdlUnique3river(open, high, low, close)

[TOP]

1.3.3.119 CdlUpsidegap2crows 0

CDLUPSIDEGAP2CROWS
Upside Gap Two Crows

Candlestick Pattern

SYNTAX	**CdlUpsidegap2crows()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Upside Gap Two Crows, The Upside Gap Two Crows is a three-day pattern.
ADDITIONAL INFO	Upside Gap Two Crows Bearish Signal: Bearish Pattern: reversal Reliability: high Identification A long white day is followed by a black candle which gaps in the direction of the trend. The final day engulfs the small black day and closes within the gap of the first two days.
EXAMPLE	CdlUpsidegap2crows()

[TOP]

1.3.3.120 CdlUpsidegap2crows 1

CDLUPSIDEGAP2CROWS
Upside Gap Two Crows

Candlestick Pattern

SYNTAX **CdlUpsidegap2crows(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Upside Gap Two Crows, The Upside Gap Two Crows is a three-day pattern.

ADDITIONAL INFO
Upside Gap Two Crows Bearish
Signal: Bearish
Pattern: reversal
Reliability: high
Identification
A long white day is followed by a black candle which gaps in the direction of the trend.
The final day engulfs the small black day and closes within the gap of the first two days.

EXAMPLE CdlUpsidegap2crows(open, high, low, close)

[TOP]

1.3.3.121 CdlXsidegap3methods 0

CDLXSIDEGAP3METHODS
Upside/Downside Gap Three Methods

Candlestick Pattern

SYNTAX **CdlXsidegap3methods()**

RETURNS NUMERIC ARRAY

DESCRIPTION Upside/Downside Gap Three Methods, the Bullish Upside Gap Three Methods Pattern, the market is in a strong bullish mood.

ADDITIONAL INFO
Upside Gap Three Methods Bullish
Signal: Bullish
Pattern: continuation
Reliability: moderate
Identification
A long white day is followed by a second long white day that gaps in the direction of the trend. The third day is black and fills the gap between the first two days.
Downside Gap Three Methods Bearish
Signal: Bearish
Pattern: continuation
Reliability: moderate
Identification
A long black day is followed by a second long black day that gaps in the direction of the trend. The third day is white and fills the gap between the first two days.

EXAMPLE CdlXsidegap3methods()

[TOP]

1.3.3.122 CdlXsidegap3methods 1

CDLXSIDEGAP3METHODS
Upside/Downside Gap Three Methods

Candlestick Pattern

SYNTAX **CdlXsidegap3methods(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Upside/Downside Gap Three Methods, the Bullish Upside Gap Three Methods Pattern, the market is in a strong bullish mood.

ADDITIONAL INFO
Upside Gap Three Methods Bullish
Signal: Bullish
Pattern: continuation
Reliability: moderate
Identification
A long white day is followed by a second long white day that gaps in the direction of the trend. The third day is black and fills the gap between the first two days.
Downside Gap Three Methods Bearish
Signal: Bearish
Pattern: continuation
Reliability: moderate
Identification
A long black day is followed by a second long black day that gaps in the direction of the trend. The third day is white and fills the gap between the first two days.

EXAMPLE CdlXsidegap3methods(open, high, low, close)

[TOP]

1.3.4 Charting

1.3.4.1 Plot

PLOT
Plot

Charting

SYNTAX	**Plot(ARRAY array, ARRAY description, ENUM color, ENUM chart type, ENUM style)**
RETURNS	
DESCRIPTION	Plot a time-series
ADDITIONAL INFO	
EXAMPLE	Plot(rsi(14), "RSI", colorBlack, chartLine, styleOwnScale)

[TOP]

1.3.4.2 Plot1

PLOT1
Plot

Charting

SYNTAX **Plot1(ARRAY array one, ARRAY array two, ARRAY description1, ARRAY description2, ENUM fillcolor, ENUM graphonecolor, ENUM graphtwocolor, ENUM chart type, ENUM style)**

RETURNS

DESCRIPTION Plot two time-series

ADDITIONAL INFO

EXAMPLE Plot1(BbandsUpper(14, 2, 2, _MaSma), BbandSlower(14, 2, 2, _MaSma), "BB UP", "BB LOW", colorRed|20|colorRed|20|0, colorTransparent, colorTransparent, chartLine, StyleSymbolNone)

[TOP]

1.3.4.3 PlotCandleStick

PLOTCANDLESTICK
Plot CandleStick

Charting

SYNTAX	**PlotCandleStick(ARRAY description, ENUM color, ENUM style)**
RETURNS	
DESCRIPTION	Plot a symbol quotes
ADDITIONAL INFO	
EXAMPLE	PlotCandleStick("CandleStick", colorBlack, StyleSymbolNone)

[TOP]

1.3.4.4 PlotCandleStick1

PLOTCANDLESTICK1
Plot CandleStick

Charting

SYNTAX **PlotCandleStick1(ARRAY open, ARRAY high, ARRAY low, ARRAY close, ARRAY description, ENUM color, ENUM style)**

RETURNS

DESCRIPTION Plot a symbol quotes

ADDITIONAL INFO

EXAMPLE PlotCandleStick1(open, high, low, close, "CandleStick", colorBlack, StyleSymbolNone)

[TOP]

1.3.4.5 PlotFixed

PLOTFIXED
Plot Fixed Graph

Charting

SYNTAX **PlotFixed(ARRAY xaxisdata, ARRAY yaxisdata, ARRAY yaxislabel, ENUM function, ARRAY description, ENUM color, ENUM chart type, ENUM style)**

RETURNS

DESCRIPTION Draw a fixed graph (see tutorial)

ADDITIONAL INFO

EXAMPLE PlotFixed(iff(GapUp(), 1, iff(GapDown(), 2, 0)), 1, "", iCount, "Number of No Gaps - Gap Up - Gap Down", colorBlack, ChartBar, StyleSymbolNone)

[TOP]

1.3.4.6 PlotSymbol

PLOTSYMBOL
Plot Symbol

Charting

SYNTAX **PlotSymbol(ARRAY array, ARRAY description, ARRAY factor, ARRAY margin, ENUM topbottom, ENUM color, ENUM border color, ENUM back color, ENUM plot symbol)**

RETURNS

DESCRIPTION Draw a symbol when the value of an element of the ARRAY is different from 0

ADDITIONAL INFO

EXAMPLE PlotSymbol(volume > 2 * sma(volume, 20), "Rise volume", 1, 1, AboveHigh, colorRed, colorBlack, colorBlack, PlotSymbolCircle)

1.3.4.7 Print

PRINT
Print To Output

Charting

SYNTAX	**Print(ARRAY text)**
RETURNS	
DESCRIPTION	Set the outpout window so it display text depending on the selected date
ADDITIONAL INFO	
EXAMPLE	Print("Hello")

[TOP]

1.3.4.8 PrintChart

PRINTCHART
Print Chart

Charting

SYNTAX **PrintChart(ARRAY string to print, ARRAY description, ENUM coordinate, ENUM color, ENUM border color, ENUM back color, ARRAY alpha)**

RETURNS

DESCRIPTION Draw a string

ADDITIONAL INFO

EXAMPLE PrintChart("Close :".close, "Close Price", TopLeft, colorRed, colorBlack, colorBlack, 255)

[TOP]

1.3.4.9 SetMaxScale

SETMAXSCALE
Set the maximum chart scale

Charting

SYNTAX	**SetMaxScale(ARRAY max)**
RETURNS	
DESCRIPTION	Set maximum chart scale
ADDITIONAL INFO	
EXAMPLE	SetMaxScale(100)

[TOP]

1.3.4.10 SetMinScale

SETMINSCALE
Set the minimum chart scale

Charting

SYNTAX	**SetMinScale(ARRAY min)**
RETURNS	
DESCRIPTION	Set minimum chart scale
ADDITIONAL INFO	
EXAMPLE	SetMinScale(0)

[TOP]

1.3.4.11 SetScale

SETSCALE
Set the chart scale

Charting

SYNTAX	**SetScale(ARRAY min, ARRAY max)**
RETURNS	
DESCRIPTION	Set chart scale
ADDITIONAL INFO	
EXAMPLE	SetScale(0, 100)

[TOP]

1.3.4.12 UpdateColor

UPDATECOLOR
Update Last Chart Color

Charting

SYNTAX	**UpdateColor(ARRAY condition, ENUM color)**
RETURNS	
DESCRIPTION	Set a different color to each bar or line
ADDITIONAL INFO	
EXAMPLE	UpdateColor(volume > 0, colorGreen)

[TOP]

1.3.5 Columns

1.3.5.1 AddColumn 0

ADDCOLUMN
Add column

Columns

SYNTAX	**AddColumn(ARRAY column name, ARRAY variable)**
RETURNS	
DESCRIPTION	Create a numeric column for the screener or the watch-list
ADDITIONAL INFO	
EXAMPLE	AddColumn("Column1", rsi(14))

[TOP]

1.3.5.2 AddColumn 1

ADDCOLUMN
Add column

Columns

SYNTAX	**AddColumn(ARRAY column name, ARRAY variable)**
RETURNS	
DESCRIPTION	Create a text column for the screener or the watch-list
ADDITIONAL INFO	
EXAMPLE	AddColumn("Column1", Market())

[TOP]

1.3.6 Database

1.3.6.1 GetData

GETDATA
Database Field

Database

SYNTAX	**GetData(STRING database name, STRING field name, ENUM type)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Get a database field data
ADDITIONAL INFO	
EXAMPLE	GetData("database_name", "field_name", LastData)

[TOP]

1.3.6.2 GetDataCount

GETDATACOUNT
Database Field Count

Database

SYNTAX	**GetDataCount(STRING database name, STRING field name)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Get the number of database elements data for each bar
ADDITIONAL INFO	
EXAMPLE	GetDataCount("database_name", "field_name")

[TOP]

1.3.6.3 GetDataCountInside

GETDATACOUNTINSIDE
Database Field Index Count Inside

Database

SYNTAX **GetDataCountInside(STRING database name, STRING field name, ARRAY low, ARRAY high, ENUM periodtype)**

RETURNS NUMERIC ARRAY

DESCRIPTION Get the number of database field elements for each bar and for the specified limits

ADDITIONAL INFO

EXAMPLE GetDataCountInside("database_name", "field_name", 0, 10, P_Hour)

[TOP]

1.3.6.4 GetDataExtern

GETDATAEXTERN
Database field for a specified symbol

Database

SYNTAX	**GetDataExtern(STRING database name, STRING field name, ARRAY type: 0 for lastdata; 1 for zero; and 2 for na, ARRAY symbol)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Get a database field data for a specified symbol
ADDITIONAL INFO	
EXAMPLE	GetDataExtern("Fundamental", "Mktcap", 0, "A")

[TOP]

1.3.6.5 GetDataIndex

GETDATAINDEX
Database Field Index

Database

SYNTAX	**GetDataIndex(STRING database name, STRING field name, ARRAY index, ENUM type)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Get an array from a database field, index parameter specify the index of the data (When using for example intraday database with an historical timeframe)
ADDITIONAL INFO	
EXAMPLE	GetDataIndex("database_name", "field_name", 0, LastData)

[TOP]

1.3.6.6 GetDataIndexDP

GETDATAINDEXDP
Database Field Index Time

Database

SYNTAX	**GetDataIndexDP(STRING database name, STRING field name, ARRAY index, ENUM periodtype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Get an array representing Year or month or day or hour... for each database field item, index parameter specify the index of the data (When using for example intraday database with an historical timeframe)
ADDITIONAL INFO	
EXAMPLE	GetDataIndexDP("database_name", "field_name", 0, P_Hour)

[TOP]

1.3.6.7 GetDataInside

GETDATAINSIDE
Database Field Index Inside

Database

SYNTAX	**GetDataInside(STRING database name, STRING field name, ARRAY low, ARRAY high, ENUM periodtype, ENUM type)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Get an array from a database field (Accept only values which date are inside the specified limits)
ADDITIONAL INFO	
EXAMPLE	GetDataInside("database_name", "field_name", 0, 10, P_Hour, LastData)

[TOP]

1.3.6.8 GetDataStringCount

GETDATASTRINGCOUNT
Database Field Count

Database

SYNTAX	**GetDataStringCount(STRING database name, STRING field name, ARRAY regular expression)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Get the number of database field elements for each bar and using the provided pattern
ADDITIONAL INFO	
EXAMPLE	GetDataStringCount("database_name", "field_name", "buy")

[TOP]

1.3.6.9 GetDataStringCountInside

GETDATASTRINGCOUNTINSIDE
Database Field Index Count Inside

Database

SYNTAX **GetDataStringCountInside(STRING database name, STRING field name, ARRAY low, ARRAY high, ENUM periodtype, ARRAY regular expression)**

RETURNS NUMERIC ARRAY

DESCRIPTION Get the number of database field elements for each bar, for the specified limits and using the provided pattern

ADDITIONAL INFO

EXAMPLE GetDataStringCountInside("database_name", "field_name", 0, 10, P_Hour, "buy")

[TOP]

1.3.6.10 IsDataExists

ISDATAEXISTS
Is Database Field Exists

Database

SYNTAX	**IsDataExists(STRING database name, STRING field name)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns 1 if the provided database field name exists, returns 0 otherwise
ADDITIONAL INFO	
EXAMPLE	IsDataExists("database_name", "field_name")

[TOP]

1.3.7 Divers

1.3.7.1 Output

OUTPUT
Set output text

Divers

SYNTAX	**Output(ARRAY text)**
RETURNS	
DESCRIPTION	Set outpout window text
ADDITIONAL INFO	
EXAMPLE	Output("Hello")

[TOP]

1.3.7.2 OutputList

OUTPUTLIST
Insert text to output list

Divers

SYNTAX	**OutputList(ARRAY text)**
RETURNS	
DESCRIPTION	Add a text to the outpout list window
ADDITIONAL INFO	
EXAMPLE	OutputList("Hello")

[TOP]

1.3.7.3 States

STATES
States

Divers

SYNTAX **States(ARRAY array, ARRAY states (ex : 70|50|30))**

RETURNS NUMERIC ARRAY

DESCRIPTION Returns an array containing different states.
Example: States(perf(close, 20), 10|0|-10)
Returns 0 if the array element value is lower then -10, returns 1 if the array element value is between -10 and 0 and returns 2 if the array element value is higher than 10

ADDITIONAL INFO

EXAMPLE States(rsi(14), 70|50|30)

[TOP]

1.3.7.4 Ticker

TICKER
Ticker

Divers

SYNTAX	**Ticker(ARRAY tickername, ENUM arrayname, ENUM type)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns an array (close, open, high, low, volume, bar) from a specified ticker
ADDITIONAL INFO	
EXAMPLE	Ticker("A", close, LastData)

[TOP]

1.3.7.5 Ticker1

TICKER1
Ticker

Divers

SYNTAX	**Ticker1(ARRAY tickername, ARRAY timeframe, ENUM arrayname, ENUM type)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns an array (close, open, high, low, volume, bar) from a specified ticker
ADDITIONAL INFO	
EXAMPLE	Ticker1("A", 1, close, LastData)

[TOP]

1.3.8 Indicators

1.3.8.1 Ad 0

AD
Chaikin A/D Line

Indicators

SYNTAX **Ad(ARRAY high, ARRAY low, ARRAY close, ARRAY volume)**

RETURNS NUMERIC ARRAY

DESCRIPTION Chaikin A/D Line

ADDITIONAL INFO The Accumulation/Distribution Line was developed by Marc Chaikin to assess the cumulative flow of money into and out of a security.
A bullish signal is given when the Accumulation/Distribution Line forms a positive divergence (for more than a month of data).
The opposite for a bearish signal.

EXAMPLE Ad(high, low, close, volume)

[TOP]

1.3.8.2 Ad 1

AD
Chaikin A/D Line

Indicators

SYNTAX	**Ad()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Chaikin A/D Line
ADDITIONAL INFO	The Accumulation/Distribution Line was developed by Marc Chaikin to assess the cumulative flow of money into and out of a security. A bullish signal is given when the Accumulation/Distribution Line forms a positive divergence (for more than a month of data). The opposite for a bearish signal.
EXAMPLE	Ad()

[TOP]

1.3.8.3 AdOsc 0

ADOSC
Chaikin A/D Oscillator

Indicators

SYNTAX **AdOsc(ARRAY high, ARRAY low, ARRAY close, ARRAY volume, ARRAY fastperiod, ARRAY slowperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION Chaikin A/D Oscillator

ADDITIONAL INFO The Chaikin Oscillator is the Moving Average Convergence Divergence indicator (MACD) applied to the Accumulation/Distribution Line.
A bullish signal happen when the price action develops a lower low into oversold zones and the oscillator diverges with a higher low and begins to rise.
A bearish signal happen when the price action develops a higher high into overbought zones and the oscillator diverges with a lower high and begins to fall.

EXAMPLE AdOsc(high, low, close, volume, 3, 10)

[TOP]

1.3.8.4 AdOsc 1

ADOSC
Chaikin A/D Oscillator

Indicators

SYNTAX **AdOsc(ARRAY fastperiod, ARRAY slowperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION Chaikin A/D Oscillator

ADDITIONAL INFO The Chaikin Oscillator is the Moving Average Convergence Divergence indicator (MACD) applied to the Accumulation/Distribution Line.
A bullish signal happen when the price action develops a lower low into oversold zones and the oscillator diverges with a higher low and begins to rise.
A bearish signal happen when the price action develops a higher high into overbought zones and the oscillator diverges with a lower high and begins to fall.

EXAMPLE AdOsc(3, 10)

[TOP]

1.3.8.5 Adx 0

ADX
Average Directional Movement Index

SYNTAX	**Adx(ARRAY high, ARRAY low, ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Average Directional Movement Index
ADDITIONAL INFO	The ADX is a Welles Wilder style moving average of the Directional Movement Index (DX). To interpret the ADX, consider a high number to be a strong trend, and a low number, a weak trend.
EXAMPLE	Adx(high, low, close, 14)

[TOP]

1.3.8.6 Adx 1

ADX
Average Directional Movement Index

Indicators

SYNTAX	**Adx(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Average Directional Movement Index
ADDITIONAL INFO	The ADX is a Welles Wilder style moving average of the Directional Movement Index (DX). To interpret the ADX, consider a high number to be a strong trend, and a low number, a weak trend.
EXAMPLE	Adx(14)

[TOP]

1.3.8.7 Adxr 0

ADXR
Average Directional Movement Index Rating

Indicators

SYNTAX	**Adxr(ARRAY high, ARRAY low, ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Average Directional Movement Index Rating
ADDITIONAL INFO	The ADXR is the average of the two ADX values. The ADXR smoothes the ADX, and is therefore less responsive, however, the ADXR filters out excessive tops and bottoms. High ADXR value is bullish while low values is bearish.
EXAMPLE	Adxr(high, low, close, 14)

[TOP]

1.3.8.8 Adxr 1

ADXR
Average Directional Movement Index Rating

Indicators

SYNTAX **Adxr(ARRAY timeperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION Average Directional Movement Index Rating

ADDITIONAL INFO The ADXR is the average of the two ADX values.
The ADXR smoothes the ADX, and is therefore less responsive, however, the ADXR filters out excessive tops and bottoms.
High ADXR value is bullish while low values is bearish.

EXAMPLE Adxr(14)

[TOP]

1.3.8.9 Apo 0

APO
Absolute Price Oscillator

Indicators

SYNTAX	**Apo(ARRAY high, ARRAY fastperiod, ARRAY slowperiod, ENUM movingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Absolute Price Oscillator
ADDITIONAL INFO	The Price Oscillator shows the difference between two moving averages. The Price Oscillator can use any time periods, unlike MACD indicator. A buy signal is generate when the Price Oscillator rises above zero, and a sell signal when the it falls below zero.
EXAMPLE	Apo(high, 3, 10, _MaSma)

[TOP]

1.3.8.10 Apo 1

APO
Absolute Price Oscillator

Indicators

SYNTAX **Apo(ARRAY fastperiod, ARRAY slowperiod, ENUM movingaveragetype)**

RETURNS NUMERIC ARRAY

DESCRIPTION Absolute Price Oscillator

ADDITIONAL INFO The Price Oscillator shows the difference between two moving averages. The Price Oscillator can use any time periods, unlike MACD indicator.
A buy signal is generate when the Price Oscillator rises above zero, and a sell signal when the it falls below zero.

EXAMPLE Apo(3, 10, _MaSma)

[TOP]

1.3.8.11 AroonDown 0

AROONDOWN
Aroon Down

Indicators

SYNTAX	**AroonDown(ARRAY high, ARRAY low, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Aroon Down
ADDITIONAL INFO	
EXAMPLE	AroonDown(high, low, 14)

[TOP]

1.3.8.12 AroonDown 1

AROONDOWN
Aroon Down

Indicators

SYNTAX	**AroonDown(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Aroon Down
ADDITIONAL INFO	
EXAMPLE	AroonDown(14)

[TOP]

1.3.8.13 AroonOsc 0

AROONOSC
Aroon Oscillator

Indicators

SYNTAX	**AroonOsc(ARRAY high, ARRAY low, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Aroon Oscillator
ADDITIONAL INFO	
EXAMPLE	AroonOsc(high, low, 14)

[TOP]

1.3.8.14 AroonOsc 1

AROONOSC
Aroon Oscillator

Indicators

SYNTAX	**AroonOsc(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Aroon Oscillator
ADDITIONAL INFO	
EXAMPLE	AroonOsc(14)

[TOP]

1.3.8.15 AroonUp 0

AROONUP
Aroon Up

Indicators

SYNTAX	**AroonUp(ARRAY high, ARRAY low, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Aroon Up
ADDITIONAL INFO	
EXAMPLE	AroonUp(high, low, 14)

1.3.8.16 AroonUp 1

AROONUP
Aroon Up

Indicators

SYNTAX	**AroonUp(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Aroon Up
ADDITIONAL INFO	
EXAMPLE	AroonUp(14)

[TOP]

1.3.8.17 Atr 0

ATR
Average True Range

SYNTAX	**Atr(ARRAY high, ARRAY low, ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Average True Range
ADDITIONAL INFO	The ATR is a measure of volatility. High ATR values indicate high volatility, and low values indicate low volatility.
EXAMPLE	Atr(high, low, close, 14)

[TOP]

1.3.8.18 Atr 1

ATR
Average True Range

Indicators

SYNTAX	**Atr(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Average True Range
ADDITIONAL INFO	The ATR is a measure of volatility. High ATR values indicate high volatility, and low values indicate low volatility.
EXAMPLE	Atr(14)

[TOP]

1.3.8.19 AvgPrice 0

AVGPRICE
Average Price

Indicators

SYNTAX	**AvgPrice(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Average Price
ADDITIONAL INFO	
EXAMPLE	AvgPrice(open, high, low, close)

[TOP]

1.3.8.20 AvgPrice 1

AVGPRICE
Average Price

Indicators

SYNTAX	**AvgPrice()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Average Price
ADDITIONAL INFO	
EXAMPLE	AvgPrice()

[TOP]

1.3.8.21 BbandSlower 0

BBANDSLOWER
Bollinger Bands (Lower band)

Indicators

SYNTAX **BbandSlower(ARRAY close, ARRAY timeperiod, ARRAY nbdevup, ARRAY nbdevdown, ENUM movingaveragetype)**

RETURNS NUMERIC ARRAY

DESCRIPTION Bollinger Bands (Lower band)

ADDITIONAL INFO Bollinger Bands are plotted at 2 standard deviations above and below a 20-day exponential moving average. As standard deviation is a measure of volatility, the bands are self-adjusting: widening during volatile markets and contracting during calmer periods.
The price are considered "overbought" when they touch the upper band and "oversold" when they touch the lower band.
Statisticly 95% of price data should fall between the two tranding bands.
A way to use Bollinger Bands is to use the upper and lower bands as price targets. If prices bounce off the lower band and cross above the 20 day average, then the upper band becomes the upper price target.
Bollinger Bands was created by John Bollinger.

EXAMPLE BbandSlower(close, 15, 2, 2, _MaSma)

[TOP]

1.3.8.22 BbandSlower 1

BBANDSLOWER
Bollinger Bands (Lower band)

Indicators

SYNTAX **BbandSlower(ARRAY timeperiod, ARRAY nbdevup, ARRAY nbdevdown, ENUM movingaveragetype)**

RETURNS NUMERIC ARRAY

DESCRIPTION Bollinger Bands (Lower band)

ADDITIONAL INFO Bollinger Bands are plotted at 2 standard deviations above and below a 20-day exponential moving average. As standard deviation is a measure of volatility, the bands are self-adjusting: widening during volatile markets and contracting during calmer periods.
The price are considered "overbought" when they touch the upper band and "oversold" when they touch the lower band.
Statisticly 95% of price data should fall between the two tranding bands.
A way to use Bollinger Bands is to use the upper and lower bands as price targets. If prices bounce off the lower band and cross above the 20 day average, then the upper band becomes the upper price target.
Bollinger Bands was created by John Bollinger.

EXAMPLE BbandSlower(14, 2, 2, _MaSma)

[TOP]

1.3.8.23 BbandsMiddle 0

BBANDSMIDDLE
Bollinger Bands (Middle band)

Indicators

SYNTAX **BbandsMiddle(ARRAY close, ARRAY timeperiod, ARRAY nbdevup, ARRAY nbdevdown, ENUM movingaveragetype)**

RETURNS NUMERIC ARRAY

DESCRIPTION Bollinger Bands (Middle band)

ADDITIONAL INFO Bollinger Bands are plotted at 2 standard deviations above and below a 20-day exponential moving average. As standard deviation is a measure of volatility, the bands are self-adjusting: widening during volatile markets and contracting during calmer periods.
The price are considered "overbought" when they touch the upper band and "oversold" when they touch the lower band.
Statisticly 95% of price data should fall between the two tranding bands.
A way to use Bollinger Bands is to use the upper and lower bands as price targets. If prices bounce off the lower band and cross above the 20 day average, then the upper band becomes the upper price target.
Bollinger Bands was created by John Bollinger.

EXAMPLE BbandsMiddle(close, 15, 2, 2, _MaSma)

[TOP]

1.3.8.24 BbandsMiddle 1

BBANDSMIDDLE
Bollinger Bands (Middle band)

Indicators

SYNTAX **BbandsMiddle(ARRAY timeperiod, ARRAY nbdevup, ARRAY nbdevdown, ENUM movingaveragetype)**

RETURNS NUMERIC ARRAY

DESCRIPTION Bollinger Bands (Middle band)

ADDITIONAL INFO Bollinger Bands are plotted at 2 standard deviations above and below a 20-day exponential moving average. As standard deviation is a measure of volatility, the bands are self-adjusting: widening during volatile markets and contracting during calmer periods.
The price are considered "overbought" when they touch the upper band and "oversold" when they touch the lower band.
Statisticly 95% of price data should fall between the two tranding bands.
A way to use Bollinger Bands is to use the upper and lower bands as price targets. If prices bounce off the lower band and cross above the 20 day average, then the upper band becomes the upper price target.
Bollinger Bands was created by John Bollinger.

EXAMPLE BbandsMiddle(14, 2, 2, _MaSma)

[TOP]

1.3.8.25 BbandsUpper 0

BBANDSUPPER
Bollinger Bands (Upper band)

Indicators

SYNTAX **BbandsUpper(ARRAY close, ARRAY timeperiod, ARRAY nbdevup, ARRAY nbdevdown, ENUM movingaveragetype)**

RETURNS NUMERIC ARRAY

DESCRIPTION Bollinger Bands (Upper band)

ADDITIONAL INFO Bollinger Bands are plotted at 2 standard deviations above and below a 20-day exponential moving average. As standard deviation is a measure of volatility, the bands are self-adjusting: widening during volatile markets and contracting during calmer periods.
The price are considered "overbought" when they touch the upper band and "oversold" when they touch the lower band.
Statisticly 95% of price data should fall between the two tranding bands.
A way to use Bollinger Bands is to use the upper and lower bands as price targets. If prices bounce off the lower band and cross above the 20 day average, then the upper band becomes the upper price target.
Bollinger Bands was created by John Bollinger.

EXAMPLE BbandsUpper(close, 15, 2, 2, _MaSma)

[TOP]

1.3.8.26 BbandsUpper 1

BBANDSUPPER
Bollinger Bands (Upper band)

Indicators

SYNTAX	**BbandsUpper(ARRAY timeperiod, ARRAY nbdevup, ARRAY nbdevdown, ENUM movingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Bollinger Bands (Upper band)
ADDITIONAL INFO	Bollinger Bands are plotted at 2 standard deviations above and below a 20-day exponential moving average. As standard deviation is a measure of volatility, the bands are self-adjusting: widening during volatile markets and contracting during calmer periods. The price are considered "overbought" when they touch the upper band and "oversold" when they touch the lower band. Statisticly 95% of price data should fall between the two tranding bands. A way to use Bollinger Bands is to use the upper and lower bands as price targets. If prices bounce off the lower band and cross above the 20 day average, then the upper band becomes the upper price target. Bollinger Bands was created by John Bollinger.
EXAMPLE	BbandsUpper(14, 2, 2, _MaSma)

[TOP]

1.3.8.27 Bop 0

BOP
Balance Of Power

Indicators

SYNTAX **Bop(ARRAY open, ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Balance Of Power

ADDITIONAL INFO The Balance of Market Power is an oscillator and supports price divergence, trends, and overbought-oversold levels. It can also help to determine market trends.
This indicator measures the velocity of the price trend.

EXAMPLE Bop(open, high, low, close)

[TOP]

1.3.8.28 Bop 1

BOP
Balance Of Power

Indicators

SYNTAX	**Bop()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Balance Of Power
ADDITIONAL INFO	The Balance of Market Power is an oscillator and supports price divergence, trends, and overbought-oversold levels. It can also help to determine market trends. This indicator measures the velocity of the price trend.
EXAMPLE	Bop()

[TOP]

1.3.8.29 Cci 0

CCI
Commodity Channel Index

Indicators

SYNTAX	**Cci(ARRAY high, ARRAY low, ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Commodity Channel Index
ADDITIONAL INFO	The CCI is designed to detect beginning and ending market trends. A price correction is expected when CCI is above +100 (overbought) and a rally when it is below -100 (oversold). You can also look for price divergence in the CCI.
EXAMPLE	Cci(high, low, close, 14)

[TOP]

1.3.8.30 Cci 1

CCI
Commodity Channel Index

Indicators

SYNTAX	**Cci(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Commodity Channel Index
ADDITIONAL INFO	The CCI is designed to detect beginning and ending market trends. A price correction is expected when CCI is above +100 (overbought) and a rally when it is below -100 (oversold). You can also look for price divergence in the CCI.
EXAMPLE	Cci(14)

[TOP]

1.3.8.31 Cmo 0

CMO
Chande Momentum Oscillator

Indicators

SYNTAX	**Cmo(ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Chande Momentum Oscillator
ADDITIONAL INFO	The Chande Momentum Indicator is a momentum indicator which use information from both up and down days. Bullish when this indicator is below level -50 and bearish when it is above level +50.
EXAMPLE	Cmo(close, 14)

[TOP]

1.3.8.32 Cmo 1

CMO
Chande Momentum Oscillator

Indicators

SYNTAX	**Cmo(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Chande Momentum Oscillator
ADDITIONAL INFO	The Chande Momentum Indicator is a momentum indicator which use information from both up and down days. Bullish when this indicator is below level -50 and bearish when it is above level +50.
EXAMPLE	Cmo(14)

[TOP]

1.3.8.33 Correl 0

CORREL
Correlation Analysis

SYNTAX	**Correl(ARRAY array, ARRAY array, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Correlation Analysis
ADDITIONAL INFO	Correlation Analysis compares two array or two samples of data to show you if one sample of data can predict the other. You can analyse coorelation between a stock against another stock or a stock against an indicator... Coorrelation between an indicator and a stock. A value above +0.7 tell you that a change in the indicator will usually predict a change in the security's price. A value below -0.7 tell you that a change in the indicator will usually predict a move of the stock price in the opposite direction. A value near 0 tell you that there is no relationship between the security's price and the indicator.
EXAMPLE	Correl(close, 0, 100)

[TOP]

1.3.8.34 Correl 1

CORREL
Correlation Analysis

Indicators

SYNTAX	**Correl(ARRAY array, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Correlation Analysis
ADDITIONAL INFO	Correlation Analysis compares two array or two samples of data to show you if one sample of data can predict the other. You can analyse coorelation between a stock against another stock or a stock against an indicator... Coorrelation between an indicator and a stock. A value above +0.7 tell you that a change in the indicator will usually predict a change in the security's price. A value below -0.7 tell you that a change in the indicator will usually predict a move of the stock price in the opposite direction. A value near 0 tell you that there is no relationship between the security's price and the indicator.
EXAMPLE	Correl(close, 100)

[TOP]

1.3.8.35 Dema 0

DEMA
Double Exponential Moving Average

Indicators

SYNTAX	**Dema(ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Double Exponential Moving Average
ADDITIONAL INFO	The DEMA is a smoothing indicator with less lag than a straight exponential moving average. DEMA is an acronym for Double Exponential Moving Average, but the calculation is more complex than just a moving average of a moving average.
EXAMPLE	Dema(close, 14)

[TOP]

1.3.8.36 Dema 1

DEMA
Double Exponential Moving Average

Indicators

SYNTAX	**Dema(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Double Exponential Moving Average
ADDITIONAL INFO	The DEMA is a smoothing indicator with less lag than a straight exponential moving average. DEMA is an acronym for Double Exponential Moving Average, but the calculation is more complex than just a moving average of a moving average.
EXAMPLE	Dema(14)

[TOP]

1.3.8.37 Dx 0

DX
Directional Movement Index

Indicators

SYNTAX **Dx(ARRAY high, ARRAY low, ARRAY close, ARRAY timeperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION Directional Movement Index

ADDITIONAL INFO Directional Movement can be used either as a system on its own or as a filter on a trend-following system.
It help trader find if a particular stock is trending or not.
Two lines are generated in a DMI study, +DI and -DI. The first line measures positive (upward) movement and the second number measures negative (downward) movement.
A buy signal is given when the +DI line crosses over the - DI line while a sell signal is generated when the +DI line crosses below the - DI line.
A low ADX value (generally less than 20) can indicate a non-trending market with low volumes whereas a cross above 20 may indicate the start of a trend (either up or down). If the ADX is over 40 and begins to fall, it can indicate the slowdown of a current trend.
This indicator can also be used to identify non-trending markets, or a deterioration of an ongoing trend.
Directional Movement Index was developed by Welles Wilder and explained in his book, New Concepts in Technical Trading Systems.

EXAMPLE Dx(high, low, close, 14)

[TOP]

1.3.8.38 Dx 1

DX
Directional Movement Index

Indicators

SYNTAX	**Dx(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Directional Movement Index
ADDITIONAL INFO	Directional Movement can be used either as a system on its own or as a filter on a trend-following system. It help trader find if a particular stock is trending or not. Two lines are generated in a DMI study, +DI and -DI. The first line measures positive (upward) movement and the second number measures negative (downward) movement. A buy signal is given when the +DI line crosses over the - DI line while a sell signal is generated when the +DI line crosses below the - DI line. A low ADX value (generally less than 20) can indicate a non-trending market with low volumes whereas a cross above 20 may indicate the start of a trend (either up or down). If the ADX is over 40 and begins to fall, it can indicate the slowdown of a current trend. This indicator can also be used to identify non-trending markets, or a deterioration of an ongoing trend. Directional Movement Index was developed by Welles Wilder and explained in his book, New Concepts in Technical Trading Systems.
EXAMPLE	Dx(14)

[TOP]

1.3.8.39 Ema 0

EMA
Exponential Moving Average

Indicators

SYNTAX	**Ema(ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Exponential Moving Average
ADDITIONAL INFO	In a Simple Moving Average, each value in the time period carries equal weight, and values outside of the time period are not included in the average. However, the Exponential Moving Average is a cumulative calculation, including all data. Past values have a diminishing contribution to the average, while more recent values have a greater contribution. This method allows the moving average to be more responsive to changes in the data.
EXAMPLE	Ema(close, 14)

[TOP]

1.3.8.40 Ema 1

EMA
Exponential Moving Average

Indicators

SYNTAX	**Ema(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Exponential Moving Average
ADDITIONAL INFO	In a Simple Moving Average, each value in the time period carries equal weight, and values outside of the time period are not included in the average. However, the Exponential Moving Average is a cumulative calculation, including all data. Past values have a diminishing contribution to the average, while more recent values have a greater contribution. This method allows the moving average to be more responsive to changes in the data.
EXAMPLE	Ema(14)

[TOP]

1.3.8.41 Fama 0

FAMA
Following Adaptive Moving Average

Indicators

SYNTAX	**Fama(ARRAY close, ARRAY fastlimit, ARRAY slowlimit)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Following Adaptive Moving Average
ADDITIONAL INFO	FAMA stands for Following Adaptive Moving Average, and is a complimentary indicator to MAMA.
EXAMPLE	Fama(close, 9, 14)

[TOP]

1.3.8.42 Fama 1

FAMA
Following Adaptive Moving Average

Indicators

SYNTAX	**Fama(ARRAY fastlimit, ARRAY slowlimit)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Following Adaptive Moving Average
ADDITIONAL INFO	FAMA stands for Following Adaptive Moving Average, and is a complimentary indicator to MAMA.
EXAMPLE	Fama(9, 14)

[TOP]

1.3.8.43 GapDown

GAPDOWN
Gap Down

Indicators

SYNTAX	**GapDown()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns 1, when a down gap happen
ADDITIONAL INFO	
EXAMPLE	GapDown()

[TOP]

1.3.8.44 GapUp

GAPUP
Gap Up

Indicators

SYNTAX	**GapUp()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns 1, when an up gap happen
ADDITIONAL INFO	
EXAMPLE	GapUp()

[TOP]

1.3.8.45 Hhv 0

HHV
Highest

Indicators

SYNTAX	**Hhv(ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Highest value over a specified period
ADDITIONAL INFO	
EXAMPLE	Hhv(close, 14)

[TOP]

1.3.8.46 Hhv 1

HHV
Highest

Indicators

SYNTAX	**Hhv(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Highest value over a specified period
ADDITIONAL INFO	
EXAMPLE	Hhv(14)

[TOP]

1.3.8.47 Ht_Dcperiod 0

HT_DCPERIOD
Hilbert Transform - Dominant Cycle Period

Indicators

SYNTAX	**Ht_Dcperiod(ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Hilbert Transform - Dominant Cycle Period
ADDITIONAL INFO	
EXAMPLE	Ht_Dcperiod(close)

[TOP]

1.3.8.48 Ht_Dcperiod 1

HT_DCPERIOD
Hilbert Transform - Dominant Cycle Period

Indicators

SYNTAX	**Ht_Dcperiod()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Hilbert Transform - Dominant Cycle Period
ADDITIONAL INFO	
EXAMPLE	Ht_Dcperiod()

[TOP]

1.3.8.49 Ht_Dcphase 0

HT_DCPHASE
Hilbert Transform - Dominant Cycle Phase

Indicators

SYNTAX **Ht_Dcphase(ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Hilbert Transform - Dominant Cycle Phase

ADDITIONAL INFO The Hilbert Transform is a technique used to generate inphase and quadrature components of a de-trended real-valued "analytic-like" signal (such as a Price Series) in order to analyze variations of the instantaneous phase and amplitude.. HTDCPhase returns the Hilbert Transform Phase of the Dominant Cycle. The Dominant Cycle Phase lies in the range of 0 to 360 degrees.
The DC Phase at a specific bar gives the phase position from 0 to 360 degrees within the current Hilbert Transform Period instantaneously measured at that bar. It is meaningful only during a cyclic period of the analytic signal waveform (price series) being measured. Its transition from 360 degrees to 0 degrees can be used to designate the start of a new cycle. It can also be utilized to signal the start or end of trending or cyclic periods. Departure from a constant rate change of phase is a sensitive way to detect the end of a cycle mode. See the examples.

EXAMPLE Ht_Dcphase(close)

[TOP]

1.3.8.50 Ht_Dcphase 1

HT_DCPHASE
Hilbert Transform - Dominant Cycle Phase

Indicators

SYNTAX **Ht_Dcphase()**

RETURNS NUMERIC ARRAY

DESCRIPTION Hilbert Transform - Dominant Cycle Phase

ADDITIONAL INFO The Hilbert Transform is a technique used to generate inphase and quadrature components of a de-trended real-valued "analytic-like" signal (such as a Price Series) in order to analyze variations of the instantaneous phase and amplitude.. HTDCPhase returns the Hilbert Transform Phase of the Dominant Cycle. The Dominant Cycle Phase lies in the range of 0 to 360 degrees.
The DC Phase at a specific bar gives the phase position from 0 to 360 degrees within the current Hilbert Transform Period instantaneously measured at that bar. It is meaningful only during a cyclic period of the analytic signal waveform (price series) being measured. Its transition from 360 degrees to 0 degrees can be used to designate the start of a new cycle. It can also be utilized to signal the start or end of trending or cyclic periods. Departure from a constant rate change of phase is a sensitive way to detect the end of a cycle mode. See the examples.

EXAMPLE Ht_Dcphase()

[TOP]

1.3.8.51 Ht_Leadsine 0

HT_LEADSINE
Hilbert Transform - SineWave

Indicators

SYNTAX **Ht_Leadsine(ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Hilbert Transform - SineWave

ADDITIONAL INFO The Hilbert Transform is a technique used to generate inphase and quadrature components of a de-trended real-valued "analytic-like" signal (such as a Price Series) in order to analyze variations of the instantaneous phase and amplitude.. HTDCPhase returns the Hilbert Transform Phase of the Dominant Cycle. The Dominant Cycle Phase lies in the range of 0 to 360 degrees. The Hilbert Transform Lead Sine is just the sine of the DC Phase advanced by 45 degrees.
The DC Phase at a specific bar gives the phase position from 0 to 360 degrees within the current Hilbert Transform Period instantaneously measured at that bar. The HTLeadSin is the sine of the DC Phase at a specific bar. It is most often used in conjunction with the HTSin indicator to identify cyclic turning points. Quoting from Market Mode Strategies.doc by John Ehlers from MESA Software, "A clear, unequivocal cycle mode indicator can be generated by plotting the Sine of the measured phase angle advanced by 45 degrees. This leading signal crosses the sinewave 1/8th of a cycle BEFORE the peaks and valleys of the cyclic turning points, enabling you to make your trading decision in time to profit from the entire amplitude swing of the cycle. A significant additional advantage is that the two indicator lines don't cross except at cyclic turning points, avoiding the false whipsaw signals of most "oscillators" when the market is in a Trend Mode. The two lines don't cross because the phase rate of change is nearly zero in a trend mode. Since the phase is not changing, the two lines separated by 45 degrees in phase never get the opportunity to cross." See the examples.

EXAMPLE Ht_Leadsine(close)

[TOP]

1.3.8.52 Ht_Leadsine 1

HT_LEADSINE
Hilbert Transform - SineWave

Indicators

SYNTAX **Ht_Leadsine()**

RETURNS NUMERIC ARRAY

DESCRIPTION Hilbert Transform - SineWave

ADDITIONAL INFO The Hilbert Transform is a technique used to generate inphase and quadrature components of a de-trended real-valued "analytic-like" signal (such as a Price Series) in order to analyze variations of the instantaneous phase and amplitude.. HTDCPhase returns the Hilbert Transform Phase of the Dominant Cycle. The Dominant Cycle Phase lies in the range of 0 to 360 degrees. The Hilbert Transform Lead Sine is just the sine of the DC Phase advanced by 45 degrees.
The DC Phase at a specific bar gives the phase position from 0 to 360 degrees within the current Hilbert Transform Period instantaneously measured at that bar. The HTLeadSin is the sine of the DC Phase at a specific bar. It is most often used in conjunction with the HTSin indicator to identify cyclic turning points. Quoting from Market Mode Strategies.doc by John Ehlers from MESA Software, "A clear, unequivocal cycle mode indicator can be generated by plotting the Sine of the measured phase angle advanced by 45 degrees. This leading signal crosses the sinewave 1/8th of a cycle BEFORE the peaks and valleys of the cyclic turning points, enabling you to make your trading decision in time to profit from the entire amplitude swing of the cycle. A significant additional advantage is that the two indicator lines don't cross except at cyclic turning points, avoiding the false whipsaw signals of most "oscillators" when the market is in a Trend Mode. The two lines don't cross because the phase rate of change is nearly zero in a trend mode. Since the phase is not changing, the two lines separated by 45 degrees in phase never get the opportunity to cross." See the examples.

EXAMPLE Ht_Leadsine()

[TOP]

1.3.8.53 Ht_Phasorphase 0

HT_PHASORPHASE
Hilbert Transform - Instantaneous Trendline

Indicators

SYNTAX	**Ht_Phasorphase(ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Hilbert Transform - Instantaneous Trendline
ADDITIONAL INFO	The Hilbert Transform is a technique used to generate inphase and quadrature components of a de-trended real-valued "analytic-like" signal (such as a Price Series) in order to analyze variations of the instantaneous phase and amplitude. HTInPhase returns the Hilbert Transform generated InPhase component of the input Price Series. The InPhase component is used in conjunction with the Quadrature component to generate the phase of the analytic signal (using the ArcTan function) at a specific bar or for the entire Price Series.
EXAMPLE	Ht_Phasorphase(close)

1.3.8.54 Ht_Phasorphase 1

HT_PHASORPHASE
Hilbert Transform - Instantaneous Trendline

Indicators

SYNTAX	**Ht_Phasorphase()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Hilbert Transform - Instantaneous Trendline
ADDITIONAL INFO	The Hilbert Transform is a technique used to generate inphase and quadrature components of a de-trended real-valued "analytic-like" signal (such as a Price Series) in order to analyze variations of the instantaneous phase and amplitude. HTInPhase returns the Hilbert Transform generated InPhase component of the input Price Series. The InPhase component is used in conjunction with the Quadrature component to generate the phase of the analytic signal (using the ArcTan function) at a specific bar or for the entire Price Series.
EXAMPLE	Ht_Phasorphase()

[TOP]

1.3.8.55 Ht_Phasorquadrature 0

HT_PHASORQUADRATURE
Hilbert Transform - Trend vs Cycle Mode

SYNTAX **Ht_Phasorquadrature(ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Hilbert Transform - Trend vs Cycle Mode

ADDITIONAL INFO The Hilbert Transform is a technique used to generate inphase and quadrature components of a de-trended real-valued "analytic-like" signal (such as a Price Series) in order to analyze variations of the instantaneous phase and amplitude. HTQuadrature returns the Hilbert Transform generated Quadrature component of the input Price Series.
The Quadrature component is used in conjunction with the InPhase component to generate the phase of the analytic signal (using the ArcTan function) at a specific bar or for the entire Price Series.

EXAMPLE Ht_Phasorquadrature(close)

[TOP]

1.3.8.56 Ht_Phasorquadrature 1

HT_PHASORQUADRATURE
Hilbert Transform - Trend vs Cycle Mode

Indicators

SYNTAX	**Ht_Phasorquadrature()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Hilbert Transform - Trend vs Cycle Mode
ADDITIONAL INFO	The Hilbert Transform is a technique used to generate inphase and quadrature components of a de-trended real-valued "analytic-like" signal (such as a Price Series) in order to analyze variations of the instantaneous phase and amplitude. HTQuadrature returns the Hilbert Transform generated Quadrature component of the input Price Series. The Quadrature component is used in conjunction with the InPhase component to generate the phase of the analytic signal (using the ArcTan function) at a specific bar or for the entire Price Series.
EXAMPLE	Ht_Phasorquadrature()

[TOP]

1.3.8.57 Ht_Sine 0

HT_SINE
Hilbert Transform - SineWave

Indicators

SYNTAX	**Ht_Sine(ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Hilbert Transform - SineWave
ADDITIONAL INFO	The Hilbert Transform is a technique used to generate inphase and quadrature components of a de-trended real-valued "analytic-like" signal (such as a Price Series) in order to analyze variations of the instantaneous phase and amplitude.. HTDCPhase returns the Hilbert Transform Phase of the Dominant Cycle. The Dominant Cycle Phase lies in the range of 0 to 360 degrees. The Hilbert Transform Sine is just the sine of the DC Phase. The DC Phase at a specific bar gives the phase position from 0 to 360 degrees within the current Hilbert Transform Period instantaneously measured at that bar. The HTSin is the sine of the DC Phase at a specific bar. It is most often used in conjunction with the HTLeadSin indicator to identify cyclic turning points. Quoting from Market Mode Strategies.doc by John Ehlers from MESA Software, "A clear, unequivocal cycle mode indicator can be generated by plotting the Sine of the measured phase angle advanced by 45 degrees. This leading signal crosses the sinewave 1/8th of a cycle BEFORE the peaks and valleys of the cyclic turning points, enabling you to make your trading decision in time to profit from the entire amplitude swing of the cycle. A significant additional advantage is that the two indicator lines don't cross except at cyclic turning points, avoiding the false whipsaw signals of most "oscillators" when the market is in a Trend Mode. The two lines don't cross because the phase rate of change is nearly zero in a trend mode. Since the phase is not changing, the two lines separated by 45 degrees in phase never get the opportunity to cross."
EXAMPLE	Ht_Sine(close)

[TOP]

1.3.8.58 Ht_Sine 1

HT_SINE
Hilbert Transform - SineWave

Indicators

SYNTAX **Ht_Sine()**

RETURNS NUMERIC ARRAY

DESCRIPTION Hilbert Transform - SineWave

ADDITIONAL INFO The Hilbert Transform is a technique used to generate inphase and quadrature components of a de-trended real-valued "analytic-like" signal (such as a Price Series) in order to analyze variations of the instantaneous phase and amplitude.. HTDCPhase returns the Hilbert Transform Phase of the Dominant Cycle. The Dominant Cycle Phase lies in the range of 0 to 360 degrees. The Hilbert Transform Sine is just the sine of the DC Phase. The DC Phase at a specific bar gives the phase position from 0 to 360 degrees within the current Hilbert Transform Period instantaneously measured at that bar. The HTSin is the sine of the DC Phase at a specific bar. It is most often used in conjunction with the HTLeadSin indicator to identify cyclic turning points. Quoting from Market Mode Strategies.doc by John Ehlers from MESA Software, "A clear, unequivocal cycle mode indicator can be generated by plotting the Sine of the measured phase angle advanced by 45 degrees. This leading signal crosses the sinewave 1/8th of a cycle BEFORE the peaks and valleys of the cyclic turning points, enabling you to make your trading decision in time to profit from the entire amplitude swing of the cycle. A significant additional advantage is that the two indicator lines don't cross except at cyclic turning points, avoiding the false whipsaw signals of most "oscillators" when the market is in a Trend Mode. The two lines don't cross because the phase rate of change is nearly zero in a trend mode. Since the phase is not changing, the two lines separated by 45 degrees in phase never get the opportunity to cross."

EXAMPLE Ht_Sine()

[TOP]

1.3.8.59 Ht_Trendline 0

HT_TRENDLINE
Hilbert Transform - Phasor Components

Indicators

SYNTAX	**Ht_Trendline(ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Hilbert Transform - Phasor Components
ADDITIONAL INFO	The Hilbert Transform is a technique used to generate inphase and quadrature components of a de-trended real-valued "analytic-like" signal (such as a Price Series) in order to analyze variations of the instantaneous phase and amplitude. HTTrendline (or MESA Instantaneous Trendline) returns the Price Series value after the Dominant Cycle of the analytic signal as generated by the Hilbert Transform has been removed. The Dominant Cycle can be thought of as being the "most likely" period (in the range of 10 to 40) of a sine function of the Price Series. The HTTrendline at a specific bar gives the current Hilbert Transform Trendline as instantaneously measured at that bar. In its Series form, the Instantaneous Trendline appears much like a Moving Average, but with minimal lag compared with the lag normally associated with such averages for equivalent periods. The HTTrendline is formed by removing the Dominant Cycle from the Price Series. See the examples.
EXAMPLE	Ht_Trendline(close)

[TOP]

1.3.8.60 Ht_Trendline 1

HT_TRENDLINE
Hilbert Transform - Phasor Components

SYNTAX **Ht_Trendline()**

RETURNS NUMERIC ARRAY

DESCRIPTION Hilbert Transform - Phasor Components

ADDITIONAL INFO The Hilbert Transform is a technique used to generate inphase and quadrature components of a de-trended real-valued "analytic-like" signal (such as a Price Series) in order to analyze variations of the instantaneous phase and amplitude. HTTrendline (or MESA Instantaneous Trendline) returns the Price Series value after the Dominant Cycle of the analytic signal as generated by the Hilbert Transform has been removed. The Dominant Cycle can be thought of as being the "most likely" period (in the range of 10 to 40) of a sine function of the Price Series.

The HTTrendline at a specific bar gives the current Hilbert Transform Trendline as instantaneously measured at that bar. In its Series form, the Instantaneous Trendline appears much like a Moving Average, but with minimal lag compared with the lag normally associated with such averages for equivalent periods. The HTTrendline is formed by removing the Dominant Cycle from the Price Series. See the examples.

EXAMPLE Ht_Trendline()

[TOP]

1.3.8.61 Ht_Trendmode 0

HT_TRENDMODE
Hilbert Transform - Phasor Components

Indicators

SYNTAX	**Ht_Trendmode(ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Hilbert Transform - Phasor Components
ADDITIONAL INFO	
EXAMPLE	Ht_Trendmode(close)

[TOP]

1.3.8.62 Ht_Trendmode 1

HT_TRENDMODE
Hilbert Transform - Phasor Components

Indicators

SYNTAX	**Ht_Trendmode()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Hilbert Transform - Phasor Components
ADDITIONAL INFO	
EXAMPLE	Ht_Trendmode()

[TOP]

1.3.8.63 Inside

INSIDE
Inside

Indicators

SYNTAX	**Inside()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns 1, when an inside day occurs
ADDITIONAL INFO	
EXAMPLE	Inside()

[TOP]

1.3.8.64 Kama 0

KAMA
Kaufman Adaptive Moving Average

Indicators

SYNTAX	**Kama(ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Kaufman Adaptive Moving Average
ADDITIONAL INFO	KAMA is an adaptive moving average, and uses the noise level of the market to determine the length of the trend required to calculate the average.
EXAMPLE	Kama(close, 14)

1.3.8.65 Kama 1

KAMA
Kaufman Adaptive Moving Average

Indicators

SYNTAX	**Kama(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Kaufman Adaptive Moving Average
ADDITIONAL INFO	KAMA is an adaptive moving average, and uses the noise level of the market to determine the length of the trend required to calculate the average.
EXAMPLE	Kama(14)

[TOP]

1.3.8.66 LinearReg 0

LINEARREG
Linear Regression

Indicators

SYNTAX	**LinearReg(ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Linear Regression
ADDITIONAL INFO	In statistics, linear regression is a regression method of modeling the conditional expected value of one variable y given the values of some other variable or variables x.
EXAMPLE	LinearReg(close, 14)

[TOP]

1.3.8.67 LinearReg 1

LINEARREG
Linear Regression

Indicators

SYNTAX	**LinearReg(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Linear Regression
ADDITIONAL INFO	In statistics, linear regression is a regression method of modeling the conditional expected value of one variable y given the values of some other variable or variables x.
EXAMPLE	LinearReg(14)

[TOP]

1.3.8.68 LinearReg_Angle 0

LINEARREG_ANGLE
Linear Regression Angle

Indicators

SYNTAX	**LinearReg_Angle(ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Linear Regression Angle
ADDITIONAL INFO	
EXAMPLE	LinearReg_Angle(close, 14)

[TOP]

1.3.8.69 LinearReg_Angle 1

LINEARREG_ANGLE
Linear Regression Angle

Indicators

SYNTAX	**LinearReg_Angle(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Linear Regression Angle
ADDITIONAL INFO	
EXAMPLE	LinearReg_Angle(14)

[TOP]

1.3.8.70 LinearReg_Intercept 0

LINEARREG_INTERCEPT
Linear Regression Intercept

Indicators

SYNTAX	**LinearReg_Intercept(ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Linear Regression Intercept
ADDITIONAL INFO	
EXAMPLE	LinearReg_Intercept(close, 14)

[TOP]

1.3.8.71 LinearReg_Intercept 1

LINEARREG_INTERCEPT
Linear Regression Intercept

Indicators

SYNTAX	**LinearReg_Intercept(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Linear Regression Intercept
ADDITIONAL INFO	
EXAMPLE	LinearReg_Intercept(14)

[TOP]

1.3.8.72 LinearReg_Slope 0

LINEARREG_SLOPE
Linear Regression Slope

Indicators

SYNTAX **LinearReg_Slope(ARRAY close, ARRAY timeperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION Linear Regression Slope

ADDITIONAL INFO Linear Regression Slope returns the slope of the Linear Regression line of the specified period.
It return the slope of a line that best fits the stock price over the specified period.
Linear Regression Slope indicator is a statistical indicator.
An up sloping Linear Regression line indicates that prices have been rising while a down sloping line indicates that prices have been falling.

EXAMPLE LinearReg_Slope(close, 14)

1.3.8.73 LinearReg_Slope 1

LINEARREG_SLOPE
Linear Regression Slope

Indicators

SYNTAX	**LinearReg_Slope(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Linear Regression Slope
ADDITIONAL INFO	Linear Regression Slope returns the slope of the Linear Regression line of the specified period. It return the slope of a line that best fits the stock price over the specified period. Linear Regression Slope indicator is a statistical indicator. An up sloping Linear Regression line indicates that prices have been rising while a down sloping line indicates that prices have been falling.
EXAMPLE	LinearReg_Slope(14)

[TOP]

1.3.8.74 Llv 0

LLV
Lowest

Indicators

SYNTAX	**Llv(ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Lowest value over a specified period
ADDITIONAL INFO	
EXAMPLE	Llv(close, 14)

[TOP]

1.3.8.75 Llv 1

LLV
Lowest

Indicators

SYNTAX	**Llv(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Lowest value over a specified period
ADDITIONAL INFO	
EXAMPLE	Llv(14)

[TOP]

1.3.8.76 Ma 0

MA
Moving Average

Indicators

SYNTAX	**Ma(ARRAY close, ARRAY timeperiod, ENUM movingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	All Moving Average
ADDITIONAL INFO	Differents type of moving average : Double Exponential Moving Average Exponential Moving Average Kaufman Adaptive Moving Average MESA Adaptive Moving Average Simple Moving Average T3 Triple Exponential Moving Average Triangular Moving Average Weighted Moving Average
EXAMPLE	Ma(close, 14, _MaSma)

[TOP]

1.3.8.77 Ma 1

MA
Moving Average

Indicators

SYNTAX	**Ma(ARRAY timeperiod, ENUM movingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Moving Average
ADDITIONAL INFO	Differents type of moving average : Double Exponential Moving Average Exponential Moving Average Kaufman Adaptive Moving Average MESA Adaptive Moving Average Simple Moving Average T3 Triple Exponential Moving Average Triangular Moving Average Weighted Moving Average
EXAMPLE	Ma(15, _MaSma)

[TOP]

1.3.8.78 Macd 0

MACD
Moving Average Convergence/Divergence

Indicators

SYNTAX **Macd(ARRAY close, ARRAY signalperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION Moving Average Convergence/Divergence

ADDITIONAL INFO MACD returns the Moving Average Convergence Divergence indicator. MACD is a momentum oscillator, yet its primary use is to trade trends.
MACD has two lines, one called MACD line or fast line and the other MACD signal or slow line.
Differents interpretation can be made using MACD.

EXAMPLE Macd(close, 15)

[TOP]

1.3.8.79 Macd 1

MACD
Moving Average Convergence/Divergence

Indicators

SYNTAX	**Macd(ARRAY signalperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Moving Average Convergence/Divergence
ADDITIONAL INFO	MACD returns the Moving Average Convergence Divergence indicator. MACD is a momentum oscillator, yet its primary use is to trade trends. MACD has two lines, one called MACD line or fast line and the other MACD signal or slow line. Differents interpretation can be made using MACD.
EXAMPLE	Macd(15)

[TOP]

1.3.8.80 MacdExt 0

MACDEXT
MACD with controllable MA type

Indicators

SYNTAX	**MacdExt(ARRAY close, ARRAY fastperiod, ENUM fastmovingaveragetype, ARRAY slowperiod, ENUM slowmovingaveragetype, ARRAY signalperiod, ENUM signalmovingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	MACD with controllable MA type
ADDITIONAL INFO	MACD returns the Moving Average Convergence Divergence indicator. MACD is a momentum oscillator, yet its primary use is to trade trends. MACD has two lines, one called MACD line or fast line and the other MACD signal or slow line. Moving Average Convergence/Divergence with controllable MA type let you choose the Moving average type to use in MACD calculations.
EXAMPLE	MacdExt(close, 5, _MaSma, 3, _MaSma, 15, _MaSma)

[TOP]

1.3.8.81 MacdExt 1

MACDEXT
MACD with controllable MA type

Indicators

SYNTAX **MacdExt(ARRAY fastperiod, ENUM fastmovingaveragetype, ARRAY slowperiod, ENUM slowmovingaveragetype, ARRAY signalperiod, ENUM signalmovingaveragetype)**

RETURNS NUMERIC ARRAY

DESCRIPTION MACD with controllable MA type

ADDITIONAL INFO MACD returns the Moving Average Convergence Divergence indicator. MACD is a momentum oscillator, yet its primary use is to trade trends.
MACD has two lines, one called MACD line or fast line and the other MACD signal or slow line.
Moving Average Convergence/Divergence with controllable MA type let you choose the Moving average type to use in MACD calculations.

EXAMPLE MacdExt(5, _MaSma, 3, _MaSma, 15, _MaSma)

[TOP]

1.3.8.82 MacdExtHist 0

MACDEXTHIST
MACD Historigramme with controllable MA type

Indicators

SYNTAX **MacdExtHist(ARRAY close, ARRAY fastperiod, ENUM fastmovingaveragetype, ARRAY slowperiod, ENUM slowmovingaveragetype, ARRAY signalperiod, ENUM signalmovingaveragetype)**

RETURNS NUMERIC ARRAY

DESCRIPTION MACD Historigramme with controllable MA type

ADDITIONAL INFO The MACD Histogram show the divergence between the MACD and its reference line (the 9-day Exponential Moving Average).
The MACD Histogram with controllable MA type let you choose the Moving average type to use in MACD Histogram calculations.

EXAMPLE MacdExtHist(close, 5, _MaSma, 3, _MaSma, 15, _MaSma)

[TOP]

1.3.8.83 MacdExtHist 1

MACDEXTHIST
MACD Historigramme with controllable MA type

Indicators

SYNTAX **MacdExtHist(ARRAY fastperiod, ENUM fastmovingaveragetype, ARRAY slowperiod, ENUM slowmovingaveragetype, ARRAY signalperiod, ENUM signalmovingaveragetype)**

RETURNS NUMERIC ARRAY

DESCRIPTION MACD Historigramme with controllable MA type

ADDITIONAL INFO The MACD Histogram show the divergence between the MACD and its reference line (the 9-day Exponential Moving Average).
The MACD Histogram with controllable MA type let you choose the Moving average type to use in MACD Histogram calculations.

EXAMPLE MacdExtHist(5, _MaSma, 3, _MaSma, 15, _MaSma)

[TOP]

1.3.8.84 MacdExtSignal 0

MACDEXTSIGNAL
MACD Signal with controllable MA type

Indicators

SYNTAX	**MacdExtSignal(ARRAY close, ARRAY fastperiod, ENUM fastmovingaveragetype, ARRAY slowperiod, ENUM slowmovingaveragetype, ARRAY signalperiod, ENUM signalmovingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	MACD Signal with controllable MA type
ADDITIONAL INFO	
EXAMPLE	MacdExtSignal(close, 5, _MaSma, 3, _MaSma, 15, _MaSma)

[TOP]

1.3.8.85 MacdExtSignal 1

MACDEXTSIGNAL
MACD Signal with controllable MA type

Indicators

SYNTAX	**MacdExtSignal(ARRAY fastperiod, ENUM fastmovingaveragetype, ARRAY slowperiod, ENUM slowmovingaveragetype, ARRAY signalperiod, ENUM signalmovingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	MACD Signal with controllable MA type
ADDITIONAL INFO	
EXAMPLE	MacdExtSignal(5, _MaSma, 3, _MaSma, 15, _MaSma)

[TOP]

1.3.8.86 MacdHist 0

MACDHIST
MACD Historigramme

Indicators

SYNTAX **MacdHist(ARRAY close, ARRAY signalperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION MACD Historigramme

ADDITIONAL INFO The MACD Histogram show the divergence between the MACD and its reference line (the 9-day Exponential Moving Average).

EXAMPLE MacdHist(close, 15)

[TOP]

1.3.8.87 MacdHist 1

MACDHIST
MACD Historigramme

Indicators

SYNTAX	**MacdHist(ARRAY signalperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	MACD Historigramme
ADDITIONAL INFO	The MACD Histogram show the divergence between the MACD and its reference line (the 9-day Exponential Moving Average).
EXAMPLE	MacdHist(15)

[TOP]

1.3.8.88 MacdSignal 0

MACDSIGNAL
MACD Signal

Indicators

SYNTAX	**MacdSignal(ARRAY close, ARRAY signalperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	MACD Signal
ADDITIONAL INFO	
EXAMPLE	MacdSignal(close, 15)

[TOP]

1.3.8.89 MacdSignal 1

MACDSIGNAL
MACD Signal

Indicators

SYNTAX	**MacdSignal(ARRAY signalperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	MACD Signal
ADDITIONAL INFO	
EXAMPLE	MacdSignal(15)

[TOP]

1.3.8.90 Mama 0

MAMA
MESA Adaptive Moving Average

Indicators

SYNTAX	**Mama(ARRAY close, ARRAY fastlimit, ARRAY slowlimit)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	MESA Adaptive Moving Average
ADDITIONAL INFO	MAMA stands for MESA Adaptive Moving Average.
EXAMPLE	Mama(close, 9, 14)

[TOP]

1.3.8.91 Mama 1

MAMA
MESA Adaptive Moving Average

Indicators

SYNTAX	**Mama(ARRAY fastlimit, ARRAY slowlimit)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	MESA Adaptive Moving Average
ADDITIONAL INFO	MAMA stands for MESA Adaptive Moving Average.
EXAMPLE	Mama(9, 14)

1.3.8.92 MedPrice 0

MEDPRICE
Median Price

Indicators

SYNTAX	**MedPrice(ARRAY high, ARRAY low)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Median Price
ADDITIONAL INFO	The Median Price function calculates the midpoint between the high and low prices for the day. The median price (called also mean or average price) provides a simplified view of the trading prices for the day.
EXAMPLE	MedPrice(high, low)

[TOP]

1.3.8.93 MedPrice 1

MEDPRICE
Median Price

Indicators

SYNTAX	**MedPrice()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Median Price
ADDITIONAL INFO	The Median Price function calculates the midpoint between the high and low prices for the day. The median price (called also mean or average price) provides a simplified view of the trading prices for the day.
EXAMPLE	MedPrice()

[TOP]

1.3.8.94 Mfi 0

MFI
Money Flow Index

Indicators

SYNTAX	**Mfi(ARRAY high, ARRAY low, ARRAY close, ARRAY volume, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Money Flow Index
ADDITIONAL INFO	The Money Flow Index is another momentum indicator illustrating the strength of money flowing into and out of a security. The essentiel difference between the Money Flow and others momentum indicator is that the money flow incorporates volume in addition to pricing information.
EXAMPLE	Mfi(high, low, close, volume, 14)

[TOP]

1.3.8.95 Mfi 1

MFI
Money Flow Index

Indicators

SYNTAX	**Mfi(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Money Flow Index
ADDITIONAL INFO	The Money Flow Index is another momentum indicator illustrating the strength of money flowing into and out of a security. The essentiel difference between the Money Flow and others momentum indicator is that the money flow incorporates volume in addition to pricing information.
EXAMPLE	Mfi(14)

[TOP]

1.3.8.96 MidPoint 0

MIDPOINT
MidPoint

Indicators

SYNTAX	**MidPoint(ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	MidPoint over period
ADDITIONAL INFO	Simply (highest value + lowest value)/2.
EXAMPLE	MidPoint(close, 14)

[TOP]

1.3.8.97 MidPoint 1

MIDPOINT
MidPoint

Indicators

SYNTAX	**MidPoint(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	MidPoint over period
ADDITIONAL INFO	Simply (highest value + lowest value)/2.
EXAMPLE	MidPoint(14)

[TOP]

1.3.8.98 MidPrice 0

MIDPRICE
MidPoint Price

Indicators

SYNTAX	**MidPrice(ARRAY high, ARRAY low, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	MidPoint Price over period
ADDITIONAL INFO	Midpoint Price over period Simply (highest high + lowest low)/2
EXAMPLE	MidPrice(high, low, 14)

[TOP]

1.3.8.99 MidPrice 1

MIDPRICE
MidPoint Price

Indicators

SYNTAX	**MidPrice(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	MidPoint Price over period
ADDITIONAL INFO	Midpoint Price over period Simply (highest high + lowest low)/2
EXAMPLE	MidPrice(14)

[TOP]

1.3.8.100 Minus_Di 0

MINUS_DI
Minus Directional Indicator

Indicators

SYNTAX	**Minus_Di(ARRAY high, ARRAY low, ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Minus Directional Indicator
ADDITIONAL INFO	
EXAMPLE	Minus_Di(high, low, close, 14)

[TOP]

1.3.8.101 Minus_Di 1

MINUS_DI
Minus Directional Indicator

Indicators

SYNTAX	**Minus_Di(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Minus Directional Indicator
ADDITIONAL INFO	
EXAMPLE	Minus_Di(14)

[TOP]

1.3.8.102 Minus_Dm 0

MINUS_DM
Minus Directional Movement

Indicators

SYNTAX	**Minus_Dm(ARRAY high, ARRAY low, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Minus Directional Movement
ADDITIONAL INFO	
EXAMPLE	Minus_Dm(high, low, 14)

[TOP]

1.3.8.103 Minus_Dm 1

MINUS_DM
Minus Directional Movement

Indicators

SYNTAX	**Minus_Dm(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Minus Directional Movement
ADDITIONAL INFO	
EXAMPLE	Minus_Dm(14)

[TOP]

1.3.8.104 Mom 0

MOM
Momentum

Indicators

SYNTAX **Mom(ARRAY close, ARRAY timeperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION Momentum

ADDITIONAL INFO The Momentum indicator provides an indication of a market's velocity
The Momentum indicator is used as a trend-following oscillator similar to the MACD
This indicator is bullish when it bottoms and turns up and bearish when it peaks and turns down.

EXAMPLE Mom(close, 14)

[TOP]

1.3.8.105 Mom 1

MOM
Momentum

Indicators

SYNTAX	**Mom(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Momentum
ADDITIONAL INFO	The Momentum indicator provides an indication of a market's velocity The Momentum indicator is used as a trend-following oscillator similar to the MACD This indicator is bullish when it bottoms and turns up and bearish when it peaks and turns down.
EXAMPLE	Mom(14)

[TOP]

1.3.8.106 Natr 0

NATR
Normalized Average True Range

SYNTAX	**Natr(ARRAY high, ARRAY low, ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Normalized Average True Range
ADDITIONAL INFO	
EXAMPLE	Natr(high, low, close, 14)

[TOP]

1.3.8.107 Natr 1

NATR
Normalized Average True Range

Indicators

SYNTAX	**Natr(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Normalized Average True Range
ADDITIONAL INFO	
EXAMPLE	Natr(14)

[TOP]

1.3.8.108 Obv 0

OBV
On Balance Volume

Indicators

SYNTAX **Obv(ARRAY close, ARRAY volume)**

RETURNS NUMERIC ARRAY

DESCRIPTION On Balance Volume

ADDITIONAL INFO On Balance Volume is a momentum indicator that relates volume to price change.
On Balance Volume keeps a running total of volume flowing into or out of a security.
When the security closes higher than the previous close, all of the day's volume is considered up-volume.
A close lower than the previous day's results in all of the day's volume considered down-volume.
Rising trends is considered bullish, while falling trends is bearish.

EXAMPLE Obv(close, volume)

[TOP]

1.3.8.109 Obv 1

OBV
On Balance Volume

Indicators

SYNTAX	**Obv()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	On Balance Volume
ADDITIONAL INFO	On Balance Volume is a momentum indicator that relates volume to price change. On Balance Volume keeps a running total of volume flowing into or out of a security. When the security closes higher than the previous close, all of the day's volume is considered up-volume. A close lower than the previous day's results in all of the day's volume considered down-volume. Rising trends is considered bullish, while falling trends is bearish.
EXAMPLE	Obv()

[TOP]

1.3.8.110 Outside

OUTSIDE
Outside

Indicators

SYNTAX	**Outside()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns 1, when an outside day occurs
ADDITIONAL INFO	
EXAMPLE	Outside()

[TOP]

1.3.8.111 Plus_Di 0

PLUS_DI
Plus Directional Indicator

Indicators

SYNTAX	**Plus_Di(ARRAY high, ARRAY low, ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Plus Directional Indicator
ADDITIONAL INFO	
EXAMPLE	Plus_Di(high, low, close, 14)

[TOP]

1.3.8.112 Plus_Di 1

PLUS_DI
Plus Directional Indicator

Indicators

SYNTAX	**Plus_Di(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Plus Directional Indicator
ADDITIONAL INFO	
EXAMPLE	Plus_Di(14)

[TOP]

1.3.8.113 Plus_Dm 0

PLUS_DM
Plus Directional Movement

Indicators

SYNTAX	**Plus_Dm(ARRAY high, ARRAY low, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Plus Directional Movement
ADDITIONAL INFO	
EXAMPLE	Plus_Dm(high, low, 14)

[TOP]

1.3.8.114 Plus_Dm 1

PLUS_DM
Plus Directional Movement

Indicators

SYNTAX	**Plus_Dm(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Plus Directional Movement
ADDITIONAL INFO	
EXAMPLE	Plus_Dm(14)

[TOP]

1.3.8.115 Ppo 0

PPO
Percentage Price Oscillator

Indicators

SYNTAX	**Ppo(ARRAY close, ARRAY fastperiod, ARRAY slowperiod, ENUM movingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Percentage Price Oscillator
ADDITIONAL INFO	The Percentage Price Oscillator indicator shows the variation in percentage among two moving averages for the price of a security. When the short-term moving average or price rises above the longer-term moving average it is considered bullish. Bearish for the opposite (Short-term moving average or price drop below the longer-term moving average).
EXAMPLE	Ppo(close, 3, 10, _MaSma)

[TOP]

1.3.8.116 Ppo 1

PPO
Percentage Price Oscillator

Indicators

SYNTAX **Ppo(ARRAY fastperiod, ARRAY slowperiod, ENUM movingaveragetype)**

RETURNS NUMERIC ARRAY

DESCRIPTION Percentage Price Oscillator

ADDITIONAL INFO The Percentage Price Oscillator indicator shows the variation in percentage among two moving averages for the price of a security.
When the short-term moving average or price rises above the longer-term moving average it is considered bullish.
Bearish for the opposite (Short-term moving average or price drop below the longer-term moving average).

EXAMPLE Ppo(3, 10, _MaSma)

[TOP]

1.3.8.117 Roc 0

ROC
Rate of change

Indicators

SYNTAX **Roc(ARRAY close, ARRAY timeperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION Rate of change

ADDITIONAL INFO The Rate of Change is an oscillator that displays the difference between the current price and the price x-time periods ago.
The Rate of change indicator show overbought/oversold areas like many others indicators.
Overbought means that the stock price is expected to fall while oversold means that the stock price is expected to fall.
But, often extremely overbought/oversold readings usually imply a continuation of the current trend.
Formula : ((price/prevPrice)-1)*100

EXAMPLE Roc(close, 14)

[TOP]

1.3.8.118 Roc 1

ROC
Rate of change

Indicators

SYNTAX	**Roc(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Rate of change
ADDITIONAL INFO	The Rate of Change is an oscillator that displays the difference between the current price and the price x-time periods ago. The Rate of change indicator show overbought/oversold areas like many others indicators. Overbought means that the stock price is expected to fall while oversold means that the stock price is expected to fall. But, often extremely overbought/oversold readings usually imply a continuation of the current trend. Formula : ((price/prevPrice)-1)*100
EXAMPLE	Roc(14)

[TOP]

1.3.8.119 Rocp 0

ROCP
Rate of change Percentage

SYNTAX **Rocp(ARRAY close, ARRAY timeperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION Rate of change Percentage

ADDITIONAL INFO The Rate of Change Percentage is an oscillator that displays the difference in percentage between the current price and the price x-time periods ago.
The Rate of change indicator show overbought/oversold areas like many others indicators.
Overbought means that the stock price is expected to fall while oversold means that the stock price is expected to fall.
But, often extremely overbought/oversold readings usually imply a continuation of the current trend.
Formula : (price-prevPrice)/prevPrice

EXAMPLE Rocp(close, 14)

[TOP]

1.3.8.120 Rocp 1

ROCP
Rate of change Percentage

Indicators

SYNTAX	**Rocp(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Rate of change Percentage
ADDITIONAL INFO	The Rate of Change Percentage is an oscillator that displays the difference in percentage between the current price and the price x-time periods ago. The Rate of change indicator show overbought/oversold areas like many others indicators. Overbought means that the stock price is expected to fall while oversold means that the stock price is expected to fall. But, often extremely overbought/oversold readings usually imply a continuation of the current trend. Formula : (price-prevPrice)/prevPrice
EXAMPLE	Rocp(14)

[TOP]

1.3.8.121 Rocr 0

ROCR
Rate of change ratio

Indicators

SYNTAX	**Rocr(ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Rate of change ratio
ADDITIONAL INFO	The Rate of Change ratio is an oscillator that displays the difference (ratio) between the current price and the price x-time periods ago. The Rate of change indicator show overbought/oversold areas like many others indicators. Overbought means that the stock price is expected to fall while oversold means that the stock price is expected to fall. But, often extremely overbought/oversold readings usually imply a continuation of the current trend. Formula : (price/prevPrice)
EXAMPLE	Rocr(close, 14)

[TOP]

1.3.8.122 Rocr 1

ROCR
Rate of change ratio

SYNTAX	**Rocr(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Rate of change ratio
ADDITIONAL INFO	The Rate of Change ratio is an oscillator that displays the difference (ratio) between the current price and the price x-time periods ago. The Rate of change indicator show overbought/oversold areas like many others indicators. Overbought means that the stock price is expected to fall while oversold means that the stock price is expected to fall. But, often extremely overbought/oversold readings usually imply a continuation of the current trend. Formula : (price/prevPrice)
EXAMPLE	Rocr(14)

[TOP]

1.3.8.123 Rocr100 0

ROCR100
Rate of change ratio (scale 100)

Indicators

SYNTAX **Rocr100(ARRAY close, ARRAY timeperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION Rate of change ratio (scale 100)

ADDITIONAL INFO The Rate of Change ratio 100 scale is an oscillator that displays the difference (in a scale of 100) between the current price and the price x-time periods ago.
The Rate of change indicator show overbought/oversold areas like many others indicators. Overbought means that the stock price is expected to fall while oversold means that the stock price is expected to fall.
But, often extremely overbought/oversold readings usually imply a continuation of the current trend.
Formula : (price/prevPrice)*100

EXAMPLE Rocr100(close, 14)

[TOP]

1.3.8.124 Rocr100 1

ROCR100
Rate of change ratio (scale 100)

Indicators

SYNTAX **Rocr100(ARRAY timeperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION Rate of change ratio (scale 100)

ADDITIONAL INFO The Rate of Change ratio 100 scale is an oscillator that displays the difference (in a scale of 100) between the current price and the price x-time periods ago.
The Rate of change indicator show overbought/oversold areas like many others indicators.
Overbought means that the stock price is expected to fall while oversold means that the stock price is expected to fall.
But, often extremely overbought/oversold readings usually imply a continuation of the current trend.
Formula : (price/prevPrice)*100

EXAMPLE Rocr100(14)

[TOP]

1.3.8.125 Rsi 0

RSI
Relative Strength Index

Indicators

SYNTAX **Rsi(ARRAY close, ARRAY timeperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION Relative Strength Index

ADDITIONAL INFO The Wilder's Relative Strength Index is a rate of change oscillator.
Relative Strength Index does not compare the relative performance of one stock to another, instead it compares the price of a stock to itself.
A reading of 70% or higher is generally an overbought position. Conversely, values near the 30% level should be considered an indication the situation has become oversold.
When the stock price is trending, use Relative Strength Index to trade strictly with the trend.
The Wilder's Relative Strength Index was developed by J. Welles Wilder, Jr.

EXAMPLE Rsi(close, 14)

[TOP]

1.3.8.126 Rsi 1

RSI
Relative Strength Index

Indicators

SYNTAX **Rsi(ARRAY timeperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION Relative Strength Index

ADDITIONAL INFO The Wilder's Relative Strength Index is a rate of change oscillator.
Relative Strength Index does not compare the relative performance of one stock to another, instead it compares the price of a stock to itself.
A reading of 70% or higher is generally an overbought position. Conversely, values near the 30% level should be considered an indication the situation has become oversold.
When the stock price is trending, use Relative Strength Index to trade strictly with the trend.
The Wilder's Relative Strength Index was developed by J. Welles Wilder, Jr.

EXAMPLE Rsi(14)

[TOP]

1.3.8.127 Sar 0

SAR
Parabolic SAR

Indicators

SYNTAX	**Sar(ARRAY high, ARRAY low, ARRAY acceleration, ARRAY maximum)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Parabolic SAR
ADDITIONAL INFO	The Parabolic SAR sets trailing price stops for long or short positions Parabolic SAR was described by Welles Wilder in his 1978 book, "New Concepts in Technical Trading Systems".
EXAMPLE	Sar(high, low, 0, 0.2)

[TOP]

1.3.8.128 Sar 1

SAR
Parabolic SAR

Indicators

SYNTAX	**Sar(ARRAY acceleration, ARRAY maximum)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Parabolic SAR
ADDITIONAL INFO	The Parabolic SAR sets trailing price stops for long or short positions Parabolic SAR was described by Welles Wilder in his 1978 book, "New Concepts in Technical Trading Systems".
EXAMPLE	Sar(0, 0.2)

[TOP]

1.3.8.129 SarExt 0

SAREXT
Parabolic SAR - Extended

Indicators

SYNTAX	**SarExt(ARRAY high, ARRAY low, ARRAY startvalue, ARRAY offsetonreverse, ARRAY accelerationinitlong, ARRAY accelerationlong, ARRAY accelerationmaxlong, ARRAY accelerationinitshort, ARRAY accelerationshort, ARRAY accelerationmaxshort)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Parabolic SAR - Extended
ADDITIONAL INFO	The Parabolic SAR sets trailing price stops for long or short positions Parabolic SAR was described by Welles Wilder in his 1978 book, "New Concepts in Technical Trading Systems".
EXAMPLE	SarExt(high, low, 0.02, 0, 0, 0, 0, 0, 0, 0)

[TOP]

1.3.8.130 SarExt 1

SAREXT
Parabolic SAR - Extended

Indicators

SYNTAX	**SarExt(ARRAY startvalue, ARRAY offsetonreverse, ARRAY accelerationinitlong, ARRAY accelerationlong, ARRAY accelerationmaxlong, ARRAY accelerationinitshort, ARRAY accelerationshort, ARRAY accelerationmaxshort)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Parabolic SAR - Extended
ADDITIONAL INFO	The Parabolic SAR sets trailing price stops for long or short positions Parabolic SAR was described by Welles Wilder in his 1978 book, "New Concepts in Technical Trading Systems".
EXAMPLE	SarExt(0.02, 0, 0, 0, 0, 0, 0, 0)

[TOP]

1.3.8.131 Sma 0

SMA
Simple Moving Average

Indicators

SYNTAX	**Sma(ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Simple Moving Average
ADDITIONAL INFO	The Simple Moving Average is calculated by summing the closing prices of the security for a period of time and then dividing this total by the number of time periods. the SMA is basically the average stock price over time. One of the moving average use is to show the trend direction of a security price.
EXAMPLE	Sma(close, 14)

[TOP]

1.3.8.132 Sma 1

SMA
Simple Moving Average

Indicators

SYNTAX **Sma(ARRAY timeperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION Simple Moving Average

ADDITIONAL INFO The Simple Moving Average is calculated by summing the closing prices of the security for a period of time and then dividing this total by the number of time periods. the SMA is basically the average stock price over time.
One of the moving average use is to show the trend direction of a security price.

EXAMPLE Sma(14)

[TOP]

1.3.8.133 Stddev 0

STDDEV
Standard Deviation

Indicators

SYNTAX	**Stddev(ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Standard Deviation
ADDITIONAL INFO	Standard Deviation is a statistical measurement of volatility. It measures how widely values range from the average value. High standard deviation means high volatility and large difference between the closing prices and the average closing price.
EXAMPLE	Stddev(close, 100)

[TOP]

1.3.8.134 Stddev 1

STDDEV
Standard Deviation

Indicators

SYNTAX	**Stddev(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Standard Deviation
ADDITIONAL INFO	Standard Deviation is a statistical measurement of volatility. It measures how widely values range from the average value. High standard deviation means high volatility and large difference between the closing prices and the average closing price.
EXAMPLE	Stddev(100)

[TOP]

1.3.8.135 StochFastd 0

STOCHFASTD
Stochastic Fast D

Indicators

SYNTAX	**StochFastd(ARRAY high, ARRAY low, ARRAY close, ARRAY fastk_period, ARRAY fastd_period, ENUM fastd_movingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Stochastic Fast D
ADDITIONAL INFO	
EXAMPLE	StochFastd(high, low, close, 5, 5, _MaSma)

1.3.8.136 StochFastd 1

STOCHFASTD
Stochastic Fast D

Indicators

SYNTAX	**StochFastd(ARRAY fastk_period, ARRAY fastd_period, ENUM fastd_movingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Stochastic Fast D
ADDITIONAL INFO	
EXAMPLE	StochFastd(5, 5, _MaSma)

[TOP]

1.3.8.137 StochFastk 0

STOCHFASTK
Stochastic Fast K

Indicators

SYNTAX	**StochFastk(ARRAY high, ARRAY low, ARRAY close, ARRAY fastk_period, ARRAY fastd_period, ENUM fastd_movingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Stochastic Fast K
ADDITIONAL INFO	
EXAMPLE	StochFastk(high, low, close, 5, 5, _MaSma)

[TOP]

1.3.8.138 StochFastk 1

STOCHFASTK
Stochastic Fast K

Indicators

SYNTAX	**StochFastk(ARRAY fastk_period, ARRAY fastd_period, ENUM fastd_movingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Stochastic Fast K
ADDITIONAL INFO	
EXAMPLE	StochFastk(5, 5, _MaSma)

[TOP]

1.3.8.139 StochRsiFastd 0

STOCHRSIFASTD
Stochastic Relative Strength Index Fast D

Indicators

SYNTAX	**StochRsiFastd(ARRAY close, ARRAY timeperiod, ARRAY fastk_period, ARRAY fastd_period, ENUM fastd_movingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Stochastic Relative Strength Index Fast D
ADDITIONAL INFO	
EXAMPLE	StochRsiFastd(close, 15, 5, 5, _MaSma)

[TOP]

1.3.8.140 StochRsiFastd 1

STOCHRSIFASTD
Stochastic Relative Strength Index Fast D

Indicators

SYNTAX	**StochRsiFastd(ARRAY timeperiod, ARRAY fastk_period, ARRAY fastd_period, ENUM fastd_movingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Stochastic Relative Strength Index Fast D
ADDITIONAL INFO	
EXAMPLE	StochRsiFastd(15, 5, 5, _MaSma)

[TOP]

1.3.8.141 StochRsiFastk 0

STOCHRSIFASTK
Stochastic Relative Strength Index Fast K

Indicators

SYNTAX	**StochRsiFastk(ARRAY close, ARRAY timeperiod, ARRAY fastk_period, ARRAY fastd_period, ENUM fastd_movingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Stochastic Relative Strength Index Fast K
ADDITIONAL INFO	
EXAMPLE	StochRsiFastk(close, 15, 5, 5, _MaSma)

[TOP]

1.3.8.142 StochRsiFastk 1

STOCHRSIFASTK
Stochastic Relative Strength Index Fast K

Indicators

SYNTAX	**StochRsiFastk(ARRAY timeperiod, ARRAY fastk_period, ARRAY fastd_period, ENUM fastd_movingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Stochastic Relative Strength Index Fast K
ADDITIONAL INFO	
EXAMPLE	StochRsiFastk(15, 5, 5, _MaSma)

[TOP]

1.3.8.143 StochSlowd 0

STOCHSLOWD
Stochastic Slow D

Indicators

SYNTAX	**StochSlowd(ARRAY high, ARRAY low, ARRAY close, ARRAY volume, ARRAY fastk_period, ARRAY slowk_period, ENUM slowk_movingaveragetype, ARRAY slowd_period, ENUM slowd_movingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Stochastic Slow D
ADDITIONAL INFO	
EXAMPLE	StochSlowd(high, low, close, volume, 5, 3, _MaSma, 3, _MaSma)

[TOP]

1.3.8.144 StochSlowd 1

STOCHSLOWD
Stochastic Slow D

Indicators

SYNTAX	**StochSlowd(ARRAY fastk_period, ARRAY slowk_period, ENUM slowk_movingaveragetype, ARRAY slowd_period, ENUM slowd_movingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Stochastic Slow D
ADDITIONAL INFO	
EXAMPLE	StochSlowd(5, 3, _MaSma, 3, _MaSma)

[TOP]

1.3.8.145 StochSlowk 0

STOCHSLOWK
Stochastic Slow K

Indicators

SYNTAX	**StochSlowk(ARRAY high, ARRAY low, ARRAY close, ARRAY volume, ARRAY fastk_period, ARRAY slowk_period, ENUM slowk_movingaveragetype, ARRAY slowd_period, ENUM slowd_movingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Stochastic Slow K
ADDITIONAL INFO	
EXAMPLE	StochSlowk(high, low, close, volume, 5, 3, _MaSma, 3, _MaSma)

[TOP]

1.3.8.146 StochSlowk 1

STOCHSLOWK
Stochastic Slow K

Indicators

SYNTAX	**StochSlowk(ARRAY fastk_period, ARRAY slowk_period, ENUM slowk_movingaveragetype, ARRAY slowd_period, ENUM slowd_movingaveragetype)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Stochastic Slow K
ADDITIONAL INFO	
EXAMPLE	StochSlowk(5, 3, _MaSma, 3, _MaSma)

[TOP]

1.3.8.147 T3 0

T3
T3

Indicators

SYNTAX **T3(ARRAY close, ARRAY timeperiod, ARRAY vfactor)**

RETURNS NUMERIC ARRAY

DESCRIPTION Triple Exponential Moving Average (T3)

ADDITIONAL INFO The T3 is a type of moving average, or smoothing function. It use the DEMA in its calculations; the DEMA which use the MA in its proper calculations.

EXAMPLE T3(close, 14, 0)

[TOP]

1.3.8.148 T3 1

T3
T3

Indicators

SYNTAX **T3(ARRAY timeperiod, ARRAY vfactor)**

RETURNS NUMERIC ARRAY

DESCRIPTION Triple Exponential Moving Average (T3)

ADDITIONAL INFO The T3 is a type of moving average, or smoothing function. It use the DEMA in its calculations; the DEMA which use the MA in its proper calculations.

EXAMPLE T3(14, 0)

[TOP]

1.3.8.149 Tema 0

TEMA
Tema

Indicators

SYNTAX	**Tema(ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Triple Exponential Moving Average
ADDITIONAL INFO	The TEMA is a smoothing indicator with less lag than a straight exponential moving average.
EXAMPLE	Tema(close, 14)

[TOP]

1.3.8.150 Tema 1

TEMA
Tema

SYNTAX	**Tema(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Triple Exponential Moving Average
ADDITIONAL INFO	The TEMA is a smoothing indicator with less lag than a straight exponential moving average.
EXAMPLE	Tema(14)

[TOP]

1.3.8.151 Trange 0

TRANGE
True Range

Indicators

SYNTAX	**Trange(ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	True Range
ADDITIONAL INFO	The True Range function is used to determine the normal trading range of a stock or commodity.
EXAMPLE	Trange(high, low, close)

[TOP]

1.3.8.152 Trange 1

TRANGE
True Range

Indicators

SYNTAX	**Trange()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	True Range
ADDITIONAL INFO	The True Range function is used to determine the normal trading range of a stock or commodity.
EXAMPLE	Trange()

[TOP]

1.3.8.153 Trima 0

TRIMA
riangular Moving Average

SYNTAX	**Trima(ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Triangular Moving Average
ADDITIONAL INFO	The Triangular Moving Average is equivalent to a Simple Moving Average of a Simple Moving Average.
EXAMPLE	Trima(close, 14)

[TOP]

1.3.8.154 Trima 1

TRIMA
Triangular Moving Average

Indicators

SYNTAX	**Trima(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Triangular Moving Average
ADDITIONAL INFO	The Triangular Moving Average is equivalent to a Simple Moving Average of a Simple Moving Average.
EXAMPLE	Trima(14)

[TOP]

1.3.8.155 Trix 0

TRIX
Trix

Indicators

SYNTAX **Trix(ARRAY close, ARRAY timeperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION 1-day Rate-Of-Change (ROC) of a Triple Smooth EMA

ADDITIONAL INFO TRIX is an indicator that correspond to a percent rate-of-change of a triple-smoothed exponential moving average of a closing price.
It is a momentum indicator which moves up over and under a zero line. A movement above is bullish, while a movement below is bearish.

EXAMPLE Trix(close, 14)

[TOP]

1.3.8.156 Trix 1

TRIX
Trix

Indicators

SYNTAX **Trix(ARRAY timeperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION 1-day Rate-Of-Change (ROC) of a Triple Smooth EMA

ADDITIONAL INFO TRIX is an indicator that correspond to a percent rate-of-change of a triple-smoothed exponential moving average of a closing price.
It is a momentum indicator which moves up over and under a zero line. A movement above is bullish, while a movement below is bearish.

EXAMPLE Trix(14)

[TOP]

1.3.8.157 Tsf 0

TSF
Time Series Forecast

Indicators

SYNTAX	**Tsf(ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Time Series Forecast
ADDITIONAL INFO	The Time Series Forecast function displays the statistical trend of a security's price over a specified time period based on linear regression analysis.
EXAMPLE	Tsf(close, 14)

[TOP]

1.3.8.158 Tsf 1

TSF
Time Series Forecast

Indicators

SYNTAX	**Tsf(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Time Series Forecast
ADDITIONAL INFO	The Time Series Forecast function displays the statistical trend of a security's price over a specified time period based on linear regression analysis.
EXAMPLE	Tsf(14)

[TOP]

1.3.8.159 TypPrice 0

TYPPRICE
Typical Price

Indicators

SYNTAX **TypPrice(ARRAY high, ARRAY low, ARRAY close)**

RETURNS NUMERIC ARRAY

DESCRIPTION Typical Price

ADDITIONAL INFO The Typical Price function calculates the average of the high, low, and closing prices for the day.
Its an interesting function as it incluse in one line information for the entire trading day.
You can also compare this line to its moving average to determine if a security is trending.

EXAMPLE TypPrice(high, low, close)

[TOP]

1.3.8.160 TypPrice 1

TYPPRICE
Typical Price

Indicators

SYNTAX **TypPrice(ARRAY timeperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION Typical Price

ADDITIONAL INFO The Typical Price function calculates the average of the high, low, and closing prices for the day.
Its an interesting function as it incluse in one line information for the entire trading day.
You can also compare this line to its moving average to determine if a security is trending.

EXAMPLE TypPrice(14)

[TOP]

1.3.8.161 UltOsc 0

ULTOSC
Ultimate Oscillator

Indicators

SYNTAX **UltOsc(ARRAY high, ARRAY low, ARRAY close, ARRAY timeperiod1, ARRAY timeperiod2, ARRAY timeperiod3)**

RETURNS NUMERIC ARRAY

DESCRIPTION Ultimate Oscillator

ADDITIONAL INFO The Ultimate Oscillator combines a stock's price action during three different time frames into one oscillator.
Oversold territory is below 30 while overbought territory is above 70.
The Ultimate Oscillator was developed by Larry Williams.

EXAMPLE UltOsc(high, low, close, 7, 14, 28)

[TOP]

1.3.8.162 UltOsc 1

ULTOSC
Ultimate Oscillator

Indicators

SYNTAX	**UltOsc(ARRAY timeperiod1, ARRAY timeperiod2, ARRAY timeperiod3)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Ultimate Oscillator
ADDITIONAL INFO	The Ultimate Oscillator combines a stock's price action during three different time frames into one oscillator. Oversold territory is below 30 while overbought territory is above 70. The Ultimate Oscillator was developed by Larry Williams.
EXAMPLE	UltOsc(7, 14, 28)

[TOP]

1.3.8.163 Var 0

VAR
Variance

Indicators

SYNTAX	**Var(ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Variance
ADDITIONAL INFO	The variance is a statistical measure of the dispersion of values arount the expected value. The standard deviation is the square root of the variance.
EXAMPLE	Var(close, 100)

[TOP]

1.3.8.164 Var 1

VAR
Variance

Indicators

SYNTAX	**Var(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Variance
ADDITIONAL INFO	The variance is a statistical measure of the dispersion of values arount the expected value. The standard deviation is the square root of the variance.
EXAMPLE	Var(100)

1.3.8.165 WclPrice 0

WCLPRICE
Weighted Close Price

Indicators

SYNTAX	**WclPrice(ARRAY high, ARRAY low, ARRAY close)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Weighted Close Price
ADDITIONAL INFO	The Weighted Close indicator calculates an average of each day's price. the Weighted Close provides a simplified view of the day.
EXAMPLE	WclPrice(high, low, close)

[TOP]

1.3.8.166 WclPrice 1

WCLPRICE
Weighted Close Price

Indicators

SYNTAX	**WclPrice()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Weighted Close Price
ADDITIONAL INFO	The Weighted Close indicator calculates an average of each day's price. the Weighted Close provides a simplified view of the day.
EXAMPLE	WclPrice()

[TOP]

1.3.8.167 Willr 0

WILLR
Williams' %R

Indicators

SYNTAX **Willr(ARRAY high, ARRAY low, ARRAY close, ARRAY timeperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION Williams' %R

ADDITIONAL INFO Williams %R is a momentum indicator that is designed to identify overbought and oversold areas in a nontrending market.
A bullish signal happen when the indicator is in an oversold area and the price change direction to the upward.
The opposite for the bearish signal.
Williams %R was developed by Larry Williams.

EXAMPLE Willr(high, low, close, 14)

1.3.8.168 Willr 1

WILLR
Williams' %R

Indicators

SYNTAX **Willr(ARRAY timeperiod)**

RETURNS NUMERIC ARRAY

DESCRIPTION Williams' %R

ADDITIONAL INFO Williams %R is a momentum indicator that is designed to identify overbought and oversold areas in a nontrending market.
A bullish signal happen when the indicator is in an oversold area and the price change direction to the upward.
The opposite for the bearish signal.
Williams %R was developed by Larry Williams.

EXAMPLE Willr(14)

[TOP]

1.3.8.169 Wma 0

WMA
Weighted Moving Average

Indicators

SYNTAX	**Wma(ARRAY close, ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Weighted Moving Average
ADDITIONAL INFO	A Weighted Moving Average is a Moving Average indicator that shows the average value of a security's price over a period of time with special emphasis on the more recent portions of the time period under analysis.
EXAMPLE	Wma(close, 14)

[TOP]

1.3.8.170 Wma 1

WMA
Weighted Moving Average

Indicators

SYNTAX	**Wma(ARRAY timeperiod)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Weighted Moving Average
ADDITIONAL INFO	A Weighted Moving Average is a Moving Average indicator that shows the average value of a security's price over a period of time with special emphasis on the more recent portions of the time period under analysis.
EXAMPLE	Wma(14)

[TOP]

1.3.9 Math

1.3.9.1 Abs

ABS
Abs

Math

SYNTAX	**Abs(ARRAY value)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the absolute value of each element in the array
ADDITIONAL INFO	
EXAMPLE	Abs(-1)

[TOP]

1.3.9.2 Atan

ATAN
Atan

Math

SYNTAX	**Atan(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the arc tangent of each element in the array
ADDITIONAL INFO	
EXAMPLE	Atan(1)

[TOP]

1.3.9.3 Avg 0

AVG
Average

Math

SYNTAX	**Avg(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculates the average of the ARRAY for all the lookback periods (current bar is included)
ADDITIONAL INFO	
EXAMPLE	Avg(close)

[TOP]

1.3.9.4 Avg 1

AVG
Average

Math

SYNTAX	**Avg(ARRAY array, ARRAY period)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculates the average of the ARRAY for the specified number of lookback periods (current bar is included)
ADDITIONAL INFO	
EXAMPLE	Avg(close, 10)

[TOP]

1.3.9.5 Ceil

CEIL
Ceiling

Math

SYNTAX	**Ceil(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	For each element in the array, calculate the lowest integer greater than or equal to the element value
ADDITIONAL INFO	
EXAMPLE	Ceil(1,2)

[TOP]

1.3.9.6 Cos

COS
Cos

Math

SYNTAX	**Cos(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the cosinus of each element in the array
ADDITIONAL INFO	
EXAMPLE	Cos(1)

[TOP]

1.3.9.7 Cosh

COSH
Cosh

Math

SYNTAX	**Cosh(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the hyperbolic cosine of each element in the array
ADDITIONAL INFO	
EXAMPLE	Cosh(1)

[TOP]

1.3.9.8 DivRem

DIVREM
DivRem

Math

SYNTAX	**DivRem(ARRAY array, ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	For each element in the array, calculate the quotion
ADDITIONAL INFO	
EXAMPLE	DivRem(1, 1)

[TOP]

1.3.9.9 Exp

EXP
Exp

Math

SYNTAX	**Exp(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	For each element in the array, calculate 'e' raised to the element value
ADDITIONAL INFO	
EXAMPLE	Exp(1)

[TOP]

1.3.9.10 Floor

FLOOR
Floor

Math

SYNTAX	**Floor(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	For each element in the array, calculate the largest integer less than or equal to the element value
ADDITIONAL INFO	
EXAMPLE	Floor(1,2)

[TOP]

1.3.9.11 Frac

FRAC
Frac

Math

SYNTAX	**Frac(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the fractional part of each element in the array
ADDITIONAL INFO	
EXAMPLE	Frac(1,2)

[TOP]

1.3.9.12 IeeeRemainder

IEEEREMAINDER
IeeeRemainder

Math

SYNTAX	**IeeeRemainder(ARRAY array, ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	For each element in the array, calculate the remainder resulting from the division of the element value by the specified number
ADDITIONAL INFO	
EXAMPLE	IeeeRemainder(1, 1)

[TOP]

1.3.9.13 Int

INT
Int

Math

SYNTAX	**Int(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the integer part of each element in the array
ADDITIONAL INFO	
EXAMPLE	Int(1,2)

[TOP]

1.3.9.14 Log

LOG
Log

Math

SYNTAX	**Log(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the logarithm of each element in the array
ADDITIONAL INFO	
EXAMPLE	Log(1)

[TOP]

1.3.9.15 Log10

LOG10
Log10

Math

SYNTAX	**Log10(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the base 10 logarithm of each element in the array
ADDITIONAL INFO	
EXAMPLE	Log10(10)

[TOP]

1.3.9.16 Max

MAX
Max

Math

SYNTAX	**Max(ARRAY array, ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	For each bar of the ARRAY, returns the largest one
ADDITIONAL INFO	
EXAMPLE	Max(perf(close), 0)

[TOP]

1.3.9.17 Min

MIN
Min

Math

SYNTAX	**Min(ARRAY array, ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	For each bar of the ARRAY, returns the smallest one
ADDITIONAL INFO	
EXAMPLE	Min(perf(close), 0)

[TOP]

1.3.9.18 Pow

POW
Pow

Math

SYNTAX	**Pow(ARRAY array, ARRAY power)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	For each element in the array, raise the element value to the specified power
ADDITIONAL INFO	
EXAMPLE	Pow(2, 2)

[TOP]

1.3.9.19 Random

RANDOM
Random Number

Math

SYNTAX	**Random(ARRAY min, ARRAY max)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns a random number between min and max
ADDITIONAL INFO	
EXAMPLE	Random(0, 100)

[TOP]

1.3.9.20 Round

ROUND
Round

Math

SYNTAX	**Round(ARRAY array, ARRAY decimals)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the rounded value of each element in the array
ADDITIONAL INFO	
EXAMPLE	Round(1.222, 2)

[TOP]

1.3.9.21 Sharpe

SHARPE
Sharpe

Math

SYNTAX	**Sharpe(ARRAY array, ARRAY period)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the sharpe ratio for the specified array and period
ADDITIONAL INFO	
EXAMPLE	Sharpe(close, 0)

[TOP]

1.3.9.22 Sign

SIGN
Sign

Math

SYNTAX	**Sign(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the sign of each element in the array
ADDITIONAL INFO	
EXAMPLE	Sign(1)

[TOP]

1.3.9.23 Sin

SIN
Sin

Math

SYNTAX	**Sin(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the sinus of each element in the array
ADDITIONAL INFO	
EXAMPLE	Sin(1)

[TOP]

1.3.9.24 Sinh

SINH
Sinh

Math

SYNTAX	**Sinh(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the hyperbolic sine of each element in the array
ADDITIONAL INFO	
EXAMPLE	Sinh(1)

[TOP]

1.3.9.25 Sqrt

SQRT
Sqrt

Math

SYNTAX	**Sqrt(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the square root of each element in the array
ADDITIONAL INFO	
EXAMPLE	Sqrt(4)

[TOP]

1.3.9.26 Sum 0

SUM
Sum

Math

SYNTAX	**Sum(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculates a cumulative sum of the ARRAY for all the lookback periods (current bar is included)
ADDITIONAL INFO	
EXAMPLE	Sum(perf(close) > 0)

[TOP]

1.3.9.27 Sum 1

SUM
Sum

Math

SYNTAX	**Sum(ARRAY array, ARRAY period)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculates a cumulative sum of the ARRAY for the specified number of lookback periods (current bar is included)
ADDITIONAL INFO	
EXAMPLE	Sum(perf(close) > 0, 20)

[TOP]

1.3.9.28 Tan

TAN
Tan

Math

SYNTAX	**Tan(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the tangent of each element in the array
ADDITIONAL INFO	
EXAMPLE	Tan(1)

[TOP]

1.3.9.29 Tanh

TANH
Tanh

Math

SYNTAX	**Tanh(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the hyperbolic tangent of each element in the array
ADDITIONAL INFO	
EXAMPLE	Tanh(1)

[TOP]

1.3.10 Optimal Signal

1.3.10.1 BSignal

BSIGNAL
Optimal signal

Optimal Signal

SYNTAX **BSignal(ARRAY min profit, ARRAY max drawdown, ARRAY minimum days, ARRAY maximum days)**

RETURNS NUMERIC ARRAY

DESCRIPTION Give you the best long and short entries and exits depending on the parameters you choosed
Example if you choose a minimum profit of 20 percent, a maximum drawdown of 10 percent,
a minimum number of bars of 10 and a maximum number of bars of 100
The system will gives you the most profitable trades that meets theses criterias
Plot the returned array in a pane that contains candelstick data to see entries and exits arrows
Click on those arrows to see additional information

ADDITIONAL INFO

EXAMPLE BSignal(30, -10, -1, 300)

[TOP]

1.3.10.2 BSignalLong

BSIGNALLONG
Optimal long signal

Optimal Signal

SYNTAX **BSignalLong(ARRAY min profit, ARRAY max drawdown, ARRAY minimum days, ARRAY maximum days)**

RETURNS NUMERIC ARRAY

DESCRIPTION Give you the best long entries and exits depending on the parameters you choosed
Example if you choose a minimum profit of 20 percent, a maximum drawdown of 10 percent,
a minimum number of bars of 10 and a maximum number of bars of 100
The system will gives you the most profitable trades that meets theses criterias
Plot the returned array in a pane that contains candelstick data to see entries and exits arrows
Click on those arrows to see additional information

ADDITIONAL INFO

EXAMPLE BSignalLong(30, -10, -1, 300)

[TOP]

1.3.10.3 BSignalShort

BSIGNALSHORT
Optimal short signal

SYNTAX **BSignalShort(ARRAY min profit, ARRAY max drawdown, ARRAY minimum days, ARRAY maximum days)**

RETURNS NUMERIC ARRAY

DESCRIPTION Give you the best short entries and exits depending on the parameters you choosed
Example if you choose a minimum profit of 20 percent, a maximum drawdown of 10 percent,
a minimum number of bars of 10 and a maximum number of bars of 100
The system will gives you the most profitable trades that meets theses criterias
Plot the returned array in a pane that contains candelstick data to see entries and exits arrows
Click on those arrows to see additional information

ADDITIONAL INFO

EXAMPLE BSignalShort(30, -10, -1, 300)

[TOP]

1.3.10.4 EntryLongProfit

ENTRYLONGPROFIT

Optimal Signal

SYNTAX **EntryLongProfit(ARRAY max drawdown, ARRAY maximum days)**

RETURNS NUMERIC ARRAY

DESCRIPTION For each bar in the CLOSE array, calculate the performance of a system that buy the current symbol at the next open bar and sell it when the system drawdown becomes higher than the specified number or the maximum number of bar is reached

ADDITIONAL INFO

EXAMPLE EntryLongProfit(-10, -1)

[TOP]

1.3.10.5 EntryLongProfitCond

ENTRYLONGPROFITCOND

Optimal Signal

SYNTAX	**EntryLongProfitCond(ARRAY condition, ARRAY max drawdown, ARRAY maximum days)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the performance of a system that buy the current symbol at the next open bar and sell it when the system drawdown becomes higher than the specified number or the maximum number of bar is reached , then move to the bar number (next bar number + last trade holding period)
ADDITIONAL INFO	
EXAMPLE	EntryLongProfitCond(1, -10, -1)

[TOP]

1.3.10.6 EntryLongProfitCondExitRule

ENTRYLONGPROFITCONDEXITRULE

Optimal Signal

SYNTAX	**EntryLongProfitCondExitRule(ARRAY condition, ARRAY exit, ARRAY max drawdown, ARRAY maximum days)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the performance of a system that buy the current symbol at the next open bar and sell it when the system drawdown becomes higher than the specified number or the maximum number of bar is reached or the exit rule is TRUE, then move to the bar number (next bar number + last trade holding period)
ADDITIONAL INFO	
EXAMPLE	EntryLongProfitCondExitRule(1, 1, -10, -1)

[TOP]

1.3.10.7 EntryLongProfitExitRule

ENTRYLONGPROFITEXITRULE

Optimal Signal

SYNTAX **EntryLongProfitExitRule(ARRAY exit, ARRAY max drawdown, ARRAY maximum days)**

RETURNS NUMERIC ARRAY

DESCRIPTION For each bar in the CLOSE array, calculate the performance of a system that buy the current symbol at the next open bar and sell it when the system drawdown becomes higher than the specified number or the maximum number of bar is reached or the exit rule is TRUE

ADDITIONAL INFO

EXAMPLE EntryLongProfitExitRule(1, -10, -1)

[TOP]

1.3.10.8 EntryLongProfitPerBar

ENTRYLONGPROFITPERBAR

Optimal Signal

SYNTAX **EntryLongProfitPerBar(ARRAY max drawdown, ARRAY maximum days)**

RETURNS NUMERIC ARRAY

DESCRIPTION For each bar in the CLOSE array, calculate the performance PER BAR of a system that buy the current symbol at the next open bar and sell it when the system drawdown becomes
higher than the specified number or the maximum number of bar is reached

ADDITIONAL INFO

EXAMPLE EntryLongProfitPerBar(-10, -1)

[TOP]

1.3.10.9 EntryLongProfitPerBarCond

ENTRYLONGPROFITPERBARCOND

Optimal Signal

SYNTAX	**EntryLongProfitPerBarCond(ARRAY condition, ARRAY max drawdown, ARRAY maximum days)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the performance PER BAR of a system that buy the current symbol at the next open bar and sell it when the system drawdown becomes higher than the specified number or the maximum number of bar is reached , then move to the bar number (next bar number + last trade holding period)
ADDITIONAL INFO	
EXAMPLE	EntryLongProfitPerBarCond(1, -10, -1)

[TOP]

1.3.10.10 EntryLongProfitPerBarCondExitRule

ENTRYLONGPROFITPERBARCONDEXITRULE

SYNTAX	**EntryLongProfitPerBarCondExitRule(ARRAY condition, ARRAY exit, ARRAY max drawdown, ARRAY maximum days)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the performance PER BAR of a system that buy the current symbol at the next open bar and sell it when the system drawdown becomes higher than the specified number or the maximum number of bar is reached or the exit rule is TRUE, then move to the bar number (next bar number + last trade holding period)
ADDITIONAL INFO	
EXAMPLE	EntryLongProfitPerBarCondExitRule(1, 1, -10, -1)

[TOP]

1.3.10.11 EntryLongProfitPerBarExitRule

ENTRYLONGPROFITPERBAREXITRULE

Optimal Signal

SYNTAX **EntryLongProfitPerBarExitRule(ARRAY exit, ARRAY max drawdown, ARRAY maximum days)**

RETURNS NUMERIC ARRAY

DESCRIPTION For each bar in the CLOSE array, calculate the performance PER BAR of a system that buy the current symbol at the next open bar and sell it when the system drawdown becomes

higher than the specified number or the maximum number of bar is reached or the exit rule is TRUE

ADDITIONAL INFO

EXAMPLE EntryLongProfitPerBarExitRule(1, -10, -1)

[TOP]

1.3.10.12 EntryShortProfit

ENTRYSHORTPROFIT

Optimal Signal

SYNTAX **EntryShortProfit(ARRAY max drawdown, ARRAY maximum days)**

RETURNS NUMERIC ARRAY

DESCRIPTION For each bar in the CLOSE array, calculate the performance of a system that short the current symbol at the next open bar and cover it when the system drawdown becomes higher than the specified number or the maximum number of bar is reached

ADDITIONAL INFO

EXAMPLE EntryShortProfit(-10, -1)

[TOP]

1.3.10.13 EntryShortProfitCond

ENTRYSHORTPROFITCOND

SYNTAX **EntryShortProfitCond(ARRAY condition, ARRAY max drawdown, ARRAY maximum days)**

RETURNS NUMERIC ARRAY

DESCRIPTION Calculate the performance of a system that short the current symbol at the next open bar and cover it when the system drawdown becomes
higher than the specified number or the maximum number of bar is reached ,
then move to the bar number (next bar number + last trade holding period)

ADDITIONAL INFO

EXAMPLE EntryShortProfitCond(1, -10, -1)

[TOP]

1.3.10.14 EntryShortProfitCondExitRule

ENTRYSHORTPROFITCONDEXITRULE

Optimal Signal

SYNTAX **EntryShortProfitCondExitRule(ARRAY condition, ARRAY exit, ARRAY max drawdown, ARRAY maximum days)**

RETURNS NUMERIC ARRAY

DESCRIPTION Calculate the performance of a system that short the current symbol at the next open bar and cover it when the system drawdown becomes
higher than the specified number or the maximum number of bar is reached or the exit rule is TRUE,
then move to the bar number (next bar number + last trade holding period)

ADDITIONAL INFO

EXAMPLE EntryShortProfitCondExitRule(1, 1, -10, -1)

[TOP]

1.3.10.15 EntryShortProfitExitRule

ENTRYSHORTPROFITEXITRULE

Optimal Signal

SYNTAX	**EntryShortProfitExitRule(ARRAY exit, ARRAY max drawdown, ARRAY maximum days)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	For each bar in the CLOSE array, calculate the performance of a system that short the current symbol at the next open bar and cover it when the system drawdown becomes higher than the specified number or the maximum number of bar is reached or the exit rule is TRUE
ADDITIONAL INFO	
EXAMPLE	EntryShortProfitExitRule(1, -10, -1)

[TOP]

1.3.10.16 EntryShortProfitPerBar

ENTRYSHORTPROFITPERBAR

Optimal Signal

SYNTAX	**EntryShortProfitPerBar(ARRAY max drawdown, ARRAY maximum days)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	For each bar in the CLOSE array, calculate the performance PER BAR of a system that short the current symbol at the next open bar and cover it when the system drawdown becomes higher than the specified number or the maximum number of bar is reached
ADDITIONAL INFO	
EXAMPLE	EntryShortProfitPerBar(-10, -1)

[TOP]

1.3.10.17 EntryShortProfitPerBarCond

ENTRYSHORTPROFITPERBARCOND

Optimal Signal

SYNTAX	**EntryShortProfitPerBarCond(ARRAY condition, ARRAY max drawdown, ARRAY maximum days)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the performance PER BAR of a system that short the current symbol at the next open bar and cover it when the system drawdown becomes higher than the specified number or the maximum number of bar is reached , then move to the bar number (next bar number + last trade holding period)
ADDITIONAL INFO	
EXAMPLE	EntryShortProfitPerBarCond(1, -10, -1)

1.3.10.18 EntryShortProfitPerBarCondExitRule

ENTRYSHORTPROFITPERBARCONDEXITRULE

SYNTAX	**EntryShortProfitPerBarCondExitRule(ARRAY condition, ARRAY exit, ARRAY max drawdown, ARRAY maximum days)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculate the performance PER BAR of a system that short the current symbol at the next open bar and cover it when the system drawdown becomes higher than the specified number or the maximum number of bar is reached or the exit rule is TRUE, then move to the bar number (next bar number + last trade holding period)
ADDITIONAL INFO	
EXAMPLE	EntryShortProfitPerBarCondExitRule(1, 1, -10, -1)

1.3.10.19 EntryShortProfitPerBarExitRule

ENTRYSHORTPROFITPERBAREXITRULE

Optimal Signal

SYNTAX **EntryShortProfitPerBarExitRule(ARRAY exit, ARRAY max drawdown, ARRAY maximum days)**

RETURNS NUMERIC ARRAY

DESCRIPTION For each bar in the CLOSE array, calculate the performance PER BAR of a system that short the current symbol at the next open bar and cover it when the system drawdown becomes
higher than the specified number or the maximum number of bar is reached or the exit rule is TRUE

ADDITIONAL INFO

EXAMPLE EntryShortProfitPerBarExitRule(1, -10, -1)

[TOP]

1.3.11 Pre-calculated-value

1.3.11.1 GetPreCalculatedValue 0

GETPRECALCULATEDVALUE
PreCalculated Value

Pre-calculated-value

SYNTAX	**GetPreCalculatedValue(STRING field)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Get a pre-calculated value
ADDITIONAL INFO	
EXAMPLE	GetPreCalculatedValue("last day volume")

[TOP]

1.3.11.2 GetPreCalculatedValue 1

GETPRECALCULATEDVALUE
PreCalculated Value

Pre-calculated-value

SYNTAX	**GetPreCalculatedValue(STRING field, ARRAY ticker name)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Get a pre-calculated value
ADDITIONAL INFO	
EXAMPLE	GetPreCalculatedValue("last day volume", "A")

[TOP]

1.3.12 Predictions

1.3.12.1 Predict

PREDICT
Predict the next value

Predictions

SYNTAX	**Predict(STRING category, STRING name)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Return an array containing the predictions of the specified model
ADDITIONAL INFO	
EXAMPLE	Predict("category_name", "perdiction_model_name")

[TOP]

1.3.12.2 PredictTicker

1.3.13 Ranking

1.3.13.1 Ranking

RANKING
Ranking System

Ranking

SYNTAX	**Ranking(STRING category, STRING name)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Apply the selected ranking system to the current symbol and returns an array containing the result for each bar
ADDITIONAL INFO	
EXAMPLE	Ranking("category_name", "ranking_system_name")

[TOP]

1.3.14 Simulator

1.3.14.1 BuyPrice

BUYPRICE Simulator
Set the buy price

SYNTAX **BuyPrice(ARRAY price, ARRAY valid for (bars))**

RETURNS

DESCRIPTION Set the buy price.
This limit order is valid for the specified number of bars, after this period the order is canceled.

ADDITIONAL INFO

EXAMPLE BuyPrice(22, close * 1.01)

[TOP]

1.3.14.2 CoverPrice

COVERPRICE
Set the cover price

Simulator

SYNTAX	**CoverPrice(ARRAY price, ARRAY exit at market after (bars))**
RETURNS	
DESCRIPTION	Set the cover price. This limit order is valid for the specified number of bars, after this period the order is canceled and an order at the market is initiated.
ADDITIONAL INFO	
EXAMPLE	CoverPrice(close * 1.01, 5)

[TOP]

1.3.14.3 Drawdown

DRAWDOWN
Drawdown

Simulator

SYNTAX	**Drawdown()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the drawdown curve of the current symbol
ADDITIONAL INFO	
EXAMPLE	Drawdown()

[TOP]

1.3.14.4 Equity

EQUITY
Equity

Simulator

SYNTAX	**Equity()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the equity curve of the current symbol, starting from 100
ADDITIONAL INFO	
EXAMPLE	Equity()

[TOP]

1.3.14.5 Optimize

OPTIMIZE
Optimize variables

Simulator

SYNTAX	**Optimize(ARRAY variable name, ARRAY min, ARRAY max, ARRAY step)**
RETURNS	
DESCRIPTION	Used by the simulator to optimize a variable
ADDITIONAL INFO	
EXAMPLE	Optimize("a", 1, 5, 1)

[TOP]

1.3.14.6 SellPrice

SELLPRICE
Set the sell price

Simulator

SYNTAX **SellPrice(ARRAY price, ARRAY exit at market after (bars))**

RETURNS

DESCRIPTION Set the sell price.
This limit order is valid for the specified number of bars, after this period the order is canceled and an order at the market is initiated.

ADDITIONAL INFO

EXAMPLE SellPrice(close * 1.01, 5)

[TOP]

1.3.14.7 SetSimCommission

SETSIMCOMMISSION
Simulation commissions

Simulator

SYNTAX	**SetSimCommission(ENUM commission type, ARRAY value)**
RETURNS	
DESCRIPTION	Update the simulator commission setting
ADDITIONAL INFO	
EXAMPLE	SetSimCommission(_Percentage, 0.02)

[TOP]

1.3.14.8 SetSimLongRank

SETSIMLONGRANK
Simulation long ranking system

Simulator

SYNTAX	**SetSimLongRank(ARRAY array)**
RETURNS	
DESCRIPTION	Override the simulator long ranking system and create a simple long simulation ranking system using the provided array
ADDITIONAL INFO	
EXAMPLE	SetSimLongRank(rsi(14))

[TOP]

1.3.14.9 SetSimPeriods

SETSIMPERIODS
Simulation periods

Simulator

SYNTAX **SetSimPeriods(ARRAY start year, ARRAY start month, ARRAY start day, ARRAY end year, ARRAY end month, ARRAY end day)**

RETURNS

DESCRIPTION Update the simulator start and end periods

ADDITIONAL INFO

EXAMPLE SetSimPeriods(2000, 1, 1, 2009, 11, 23)

[TOP]

1.3.14.10 SetSimRefSymbol

SETSIMREFSYMBOL
Simulation reference symbol

Simulator

SYNTAX	**SetSimRefSymbol(ARRAY symbol name)**
RETURNS	
DESCRIPTION	Update the simulator reference symbol
ADDITIONAL INFO	
EXAMPLE	SetSimRefSymbol("^RUT")

[TOP]

1.3.14.11 SetSimSetting

SETSIMSETTING
Simulation settings

Simulator

SYNTAX **SetSimSetting(ENUM setting item, ARRAY value)**

RETURNS

DESCRIPTION Update the simulator settings

ADDITIONAL INFO

EXAMPLE SetSimSetting(_NbPositions, 10)

[TOP]

1.3.14.12 SetSimShortRank

SETSIMSHORTRANK
Simulation short ranking system

Simulator

SYNTAX **SetSimShortRank(ARRAY array)**

RETURNS

DESCRIPTION Override the simulator short ranking system and create a simple short simulation ranking system using the provided array

ADDITIONAL INFO

EXAMPLE SetSimShortRank(1 / rsi(14))

[TOP]

1.3.14.13 SetSimStop

SETSIMSTOP
Simulation stops

Simulator

SYNTAX	**SetSimStop(ENUM stop type, ENUM stop entry, ARRAY value, ARRAY reentry)**
RETURNS	
DESCRIPTION	Update the simulator stops
ADDITIONAL INFO	
EXAMPLE	SetSimStop(_StopLoss, _Percent, 20, 1)

[TOP]

1.3.14.14 SetSimTiming

SETSIMTIMING
Simulation timing

Simulator

SYNTAX	**SetSimTiming(ENUM entry type, ENUM entry price, ARRAY decalage)**
RETURNS	
DESCRIPTION	Update the simulator entries and exits timing
ADDITIONAL INFO	
EXAMPLE	SetSimTiming(_Buy, _Open, 1)

[TOP]

1.3.14.15 ShortPrice

SHORTPRICE
Set the short price

Simulator

SYNTAX **ShortPrice(ARRAY price, ARRAY valid for (bars))**

RETURNS

DESCRIPTION Set the short price.
This limit order is valid for the specified number of bars, after this period the order is canceled.

ADDITIONAL INFO

EXAMPLE ShortPrice(22, close * 1.01)

[TOP]

1.3.15 String

1.3.15.1 GetDataString

GETDATASTRING
String Database Field

String

SYNTAX	**GetDataString(STRING database name, STRING field name, ARRAY regular expression)**
RETURNS	TEXT ARRAY
DESCRIPTION	Get a string array from a database field
ADDITIONAL INFO	
EXAMPLE	GetDataString("database_name", "field_name", "buy")

[TOP]

1.3.15.2 GetDataStringIndex

GETDATASTRINGINDEX
String Database Field Index

String

SYNTAX **GetDataStringIndex(STRING database name, STRING field name, ARRAY index, ARRAY regular expression)**

RETURNS TEXT ARRAY

DESCRIPTION Get a string array from a database field, index parameter specify the index of the data (When using for example intraday database with an historical timeframe)

ADDITIONAL INFO

EXAMPLE GetDataStringIndex("database_name", "field_name", 0, "buy")

[TOP]

1.3.15.3 GetDataStringInside

GETDATASTRINGINSIDE
String Database Field Index Inside

String

SYNTAX	**GetDataStringInside(STRING database name, STRING field name, ARRAY low, ARRAY high, ENUM periodtype, ARRAY regular expression)**
RETURNS	TEXT ARRAY
DESCRIPTION	Get a string array from a database field (Accept only values which date are inside the specified limits)
ADDITIONAL INFO	
EXAMPLE	GetDataStringInside("database_name", "field_name", 0, 10, P_Hour, "buy")

[TOP]

1.3.15.4 StringContains

STRINGCONTAINS
String Contains

String

SYNTAX	**StringContains(ARRAY string, ARRAY string)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns 1 if the first string parameter contains the second string parameter, 0 otherwise
ADDITIONAL INFO	
EXAMPLE	StringContains("HELLO", "HE")

[TOP]

1.3.15.5 StringEqual

STRINGEQUAL
String Equals

String

SYNTAX	**StringEqual(ARRAY string, ARRAY string)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns 1 if the two strings are equal, 0 otherwise
ADDITIONAL INFO	
EXAMPLE	StringEqual("HELLO", "HELLO")

[TOP]

1.3.15.6 StringExtract

STRINGEXTRACT
String Extract

String

SYNTAX **StringExtract(ARRAY string, ARRAY value, ARRAY value)**

RETURNS TEXT ARRAY

DESCRIPTION Extract a string from the string in the first parameter, starting at the specified number in the second parameter and ending at the specified number in the third parameter

ADDITIONAL INFO

EXAMPLE StringExtract("HELLO", 1, 3)

[TOP]

1.3.15.7 StringExtractEnds

STRINGEXTRACTENDS
String Extract Ends

String

SYNTAX	**StringExtractEnds(ARRAY string, ARRAY value)**
RETURNS	TEXT ARRAY
DESCRIPTION	Extract a string from the string in the first parameter, starting from (Length of the first parameter string - number in the second parameter) and ending at the length of the first parameter string
ADDITIONAL INFO	
EXAMPLE	StringExtractEnds("HELLO", 2)

[TOP]

1.3.15.8 StringExtractStart

STRINGEXTRACTSTART
String Extract Start

String

SYNTAX	**StringExtractStart(ARRAY string, ARRAY value)**
RETURNS	TEXT ARRAY
DESCRIPTION	Extract a string from the string in the first parameter, starting at 0 and ending at the specified number in the second parameter
ADDITIONAL INFO	
EXAMPLE	StringExtractStart("HELLO", 2)

[TOP]

1.3.15.9 StringInsert

STRINGINSERT String Insert	String

SYNTAX	**StringInsert(ARRAY string, ARRAY toinsert, ARRAY index)**
RETURNS	TEXT ARRAY
DESCRIPTION	Insert in the first parameter the second parameter string at the specified index number
ADDITIONAL INFO	
EXAMPLE	StringInsert(" EVERYBODY", "HELLO", 0)

[TOP]

1.3.15.10 StringLength

STRINGLENGTH
String Length

String

SYNTAX	**StringLength(ARRAY string)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the length of the string
ADDITIONAL INFO	
EXAMPLE	StringLength("HELLO")

[TOP]

1.3.15.11 StringReplace

STRINGREPLACE
String Replace

String

SYNTAX	**StringReplace(ARRAY string, ARRAY oldvalue, ARRAY newvalue)**
RETURNS	TEXT ARRAY
DESCRIPTION	Replace in the first parameter string, oldValue with newValue
ADDITIONAL INFO	
EXAMPLE	StringReplace("HELLO", "O", "OOOO")

[TOP]

1.3.16 Symbol Info

1.3.16.1 Address

ADDRESS
Address

Symbol Info

SYNTAX	**Address()**
RETURNS	TEXT ARRAY
DESCRIPTION	Returns the symbol's Address
ADDITIONAL INFO	
EXAMPLE	Address()

[TOP]

1.3.16.2 Country

COUNTRY
Country

Symbol Info

SYNTAX	**Country()**
RETURNS	TEXT ARRAY
DESCRIPTION	Returns the symbol's Country
ADDITIONAL INFO	
EXAMPLE	Country()

[TOP]

1.3.16.3 Currency

CURRENCY
Currency

Symbol Info

SYNTAX	**Currency()**
RETURNS	TEXT ARRAY
DESCRIPTION	Returns the symbol's Currency
ADDITIONAL INFO	
EXAMPLE	Currency()

[TOP]

1.3.16.4 FullName

FULLNAME
FullName

Symbol Info

SYNTAX	**FullName()**
RETURNS	TEXT ARRAY
DESCRIPTION	Returns the symbol's FullName
ADDITIONAL INFO	
EXAMPLE	FullName()

[TOP]

1.3.16.5 GetPreCalculatedValueString 0

1.3.16.6 GetPreCalculatedValueString 1

1.3.16.7 Group

GROUP
Group

Symbol Info

SYNTAX	**Group()**
RETURNS	TEXT ARRAY
DESCRIPTION	Returns the symbol's Group
ADDITIONAL INFO	
EXAMPLE	Group()

[TOP]

1.3.16.8 Index

INDEX
Index

Symbol Info

SYNTAX	**Index()**
RETURNS	TEXT ARRAY
DESCRIPTION	Returns the symbol's Index
ADDITIONAL INFO	
EXAMPLE	Index()

[TOP]

1.3.16.9 Industry

INDUSTRY
Industry

Symbol Info

SYNTAX	**Industry()**
RETURNS	TEXT ARRAY
DESCRIPTION	Returns the symbol's Industry
ADDITIONAL INFO	
EXAMPLE	Industry()

[TOP]

1.3.16.10 Market

MARKET
Market

Symbol Info

SYNTAX	**Market()**
RETURNS	TEXT ARRAY
DESCRIPTION	Returns the symbol's Market
ADDITIONAL INFO	
EXAMPLE	Market()

1.3.16.11 Name

NAME
Name

SYNTAX	**Name()**
RETURNS	TEXT ARRAY
DESCRIPTION	Returns the symbol's Name
ADDITIONAL INFO	
EXAMPLE	Name()

[TOP]

1.3.16.12 Name1

NAME1
Name1

Symbol Info

SYNTAX	**Name1()**
RETURNS	TEXT ARRAY
DESCRIPTION	Returns the symbol's Name1
ADDITIONAL INFO	
EXAMPLE	Name1()

[TOP]

1.3.16.13 Name2

NAME2
Name2

Symbol Info

SYNTAX	**Name2()**
RETURNS	TEXT ARRAY
DESCRIPTION	Returns the symbol's Name2
ADDITIONAL INFO	
EXAMPLE	Name2()

[TOP]

1.3.16.14 Name3

NAME3
Name3

SYNTAX	**Name3()**
RETURNS	TEXT ARRAY
DESCRIPTION	Returns the symbol's Name3
ADDITIONAL INFO	
EXAMPLE	Name3()

[TOP]

1.3.16.15 Sector

SECTOR
Sector

SYNTAX	**Sector()**
RETURNS	TEXT ARRAY
DESCRIPTION	Returns the symbol's Sector
ADDITIONAL INFO	
EXAMPLE	Sector()

[TOP]

1.3.16.16 Website

WEBSITE
Website

Symbol Info

SYNTAX	**Website()**
RETURNS	TEXT ARRAY
DESCRIPTION	Returns the symbol's Website
ADDITIONAL INFO	
EXAMPLE	Website()

[TOP]

1.3.17 Technical

1.3.17.1 AvgIf 0

AVGIF
Average If

Technical

SYNTAX	**AvgIf(ARRAY condition, ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculates, when condition is TRUE, the average of the ARRAY for all the lookback periods (current bar is included)
ADDITIONAL INFO	
EXAMPLE	AvgIf(perf(close) > 0, close)

[TOP]

1.3.17.2 AvgIf 1

AVGIF
Average If

Technical

SYNTAX	**AvgIf(ARRAY condition, ARRAY array, ARRAY period)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculates, when condition is TRUE, the average of the ARRAY for the specified number of lookback periods (current bar is included) (When the condition is true)
ADDITIONAL INFO	
EXAMPLE	AvgIf(perf(close) > 0, close, 10)

[TOP]

1.3.17.3 BarsSince

BARSSINCE
Bars Since

Technical

SYNTAX	**BarsSince(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculates the number of bars that have passed since the array was different from 0
ADDITIONAL INFO	
EXAMPLE	BarsSince(perf(close) > 10)

[TOP]

1.3.17.4 Count

COUNT
Count

Technical

SYNTAX	**Count()**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the number of bars
ADDITIONAL INFO	
EXAMPLE	Count()

[TOP]

1.3.17.5 Cross

CROSS
Cross

Technical

SYNTAX	**Cross(ARRAY , ARRAY)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns '1', when the first array crosses above the second array, otherwise the result is '0'
ADDITIONAL INFO	
EXAMPLE	Cross(sma(14), sma(21))

[TOP]

1.3.17.6 GFun

GFUN
Gaussian

Technical

SYNTAX	**GFun(ARRAY array, ARRAY period)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the gaussian for the specified array
ADDITIONAL INFO	
EXAMPLE	GFun(volume, 30)

[TOP]

1.3.17.7 HhvLb 0

HHVLB
Highest Since

Technical

SYNTAX **HhvLb(ARRAY array)**

RETURNS NUMERIC ARRAY

DESCRIPTION Returns the number of bars since the array reached its highest value

ADDITIONAL INFO

EXAMPLE HhvLb(close)

[TOP]

1.3.17.8 HhvLb 1

HHVLB
Highest Since

Technical

SYNTAX	**HhvLb(ARRAY array, ARRAY period)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the number of bars since the array reached its highest value over the specified period
ADDITIONAL INFO	
EXAMPLE	HhvLb(close, 20)

[TOP]

1.3.17.9 Iff

IFF
IF

Technical

SYNTAX	**Iff(ARRAY , ARRAY , ARRAY)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	A conditional function that returns the value of the first parameter if condition is true (different from 0), if condition is false then it returns the value of the second parameter.
ADDITIONAL INFO	
EXAMPLE	Iff(rsi(14) > 50, 1, 0)

[TOP]

1.3.17.10 IsFalse

ISFALSE
Is False

Technical

SYNTAX	**IsFalse(ARRAY array, ARRAY lag)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns 1 if all the values between the current bar and the lag bar are inferior or equal to 0
ADDITIONAL INFO	
EXAMPLE	IsFalse(perf(close, 10), 20)

[TOP]

1.3.17.11 IsNa

ISNA
Is NA

Technical

SYNTAX	**IsNa(ARRAY value)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns 1 if all elements in the array equal to NA
ADDITIONAL INFO	
EXAMPLE	IsNa(1)

[TOP]

1.3.17.12 IsNaN

ISNAN
Is NaN

Technical

SYNTAX	**IsNaN(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns 1 if all elements in the array equal to NaN
ADDITIONAL INFO	
EXAMPLE	IsNaN(1)

[TOP]

1.3.17.13 IsNoNa

ISNONA
Is not NA

Technical

SYNTAX	**IsNoNa(ARRAY value)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns 0 if at least one elements in the array is equal to NA
ADDITIONAL INFO	
EXAMPLE	IsNoNa(1)

[TOP]

1.3.17.14 IsNoNaN

ISNONAN
Is not NaN

Technical

SYNTAX	**IsNoNaN(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns 0 if at least one elements in the array is equal to NaN
ADDITIONAL INFO	
EXAMPLE	IsNoNaN(1)

[TOP]

1.3.17.15 IsNoNull

ISNONULL
Is not NULL

Technical

SYNTAX **IsNoNull(ARRAY value)**

RETURNS NUMERIC ARRAY

DESCRIPTION Returns 0 if at least one elements in the array is equal to 0

ADDITIONAL INFO

EXAMPLE IsNoNull(1)

[TOP]

1.3.17.16 IsNoZero

ISNOZERO
Is not Zero

Technical

SYNTAX **IsNoZero(ARRAY array)**

RETURNS NUMERIC ARRAY

DESCRIPTION Returns 0 if at least one elements in the array is equal to 0

ADDITIONAL INFO

EXAMPLE IsNoZero(1)

[TOP]

1.3.17.17 IsNull

ISNULL
Is NULL

Technical

SYNTAX	**IsNull(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns 1 if all elements in the array equal to 0
ADDITIONAL INFO	
EXAMPLE	IsNull(1)

[TOP]

1.3.17.18 IsTrue

ISTRUE
Is True

Technical

SYNTAX	**IsTrue(ARRAY array, ARRAY lag)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns 1 if all the values between the current bar and the lag bar are superior to 0
ADDITIONAL INFO	
EXAMPLE	IsTrue(perf(close, 10), 20)

[TOP]

1.3.17.19 LastNotNullValue

LASTNOTNULLVALUE Last not null		Technical

SYNTAX	**LastNotNullValue(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the last (not equal to 0) value of the ARRAY
ADDITIONAL INFO	
EXAMPLE	LastNotNullValue(close)

[TOP]

1.3.17.20 LastValue

LASTVALUE
Last Value

Technical

SYNTAX	**LastValue(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the last value of the ARRAY
ADDITIONAL INFO	
EXAMPLE	LastValue(close)

[TOP]

1.3.17.21 LlvLb 0

LLVLB
Lowest Since

Technical

SYNTAX	**LlvLb(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the number of bars since the array reached its lowest value
ADDITIONAL INFO	
EXAMPLE	LlvLb(close)

[TOP]

1.3.17.22 LlvLb 1

LLVLB
Lowest Since

Technical

SYNTAX	**LlvLb(ARRAY array, ARRAY period)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the number of bars since the array reached its lowest value over the specified period
ADDITIONAL INFO	
EXAMPLE	LlvLb(close, 20)

[TOP]

1.3.17.23 NaNtoLast

NANTOLAST
NaN to Last

Technical

SYNTAX	**NaNtoLast(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Update the NA values with the last non NaN values
ADDITIONAL INFO	
EXAMPLE	NaNtoLast(perf(close, -10))

[TOP]

1.3.17.24 NaNtoZero

NANTOZERO
NaN values to Zero

Technical

SYNTAX **NaNtoZero(ARRAY array, ARRAY type)**

RETURNS NUMERIC ARRAY

DESCRIPTION Change NaN value to Zero in the specified array.
Set zero to the parameter 'type' to update only the firsts NaN, Set one to update all NaN values

ADDITIONAL INFO

EXAMPLE NaNtoZero(rsi(14), 0)

[TOP]

1.3.17.25 Perf

PERF
Performance over a specified period

Technical

SYNTAX	**Perf(ARRAY array, ARRAY period)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the performance of the array for the specified period (k-step Return)
ADDITIONAL INFO	
EXAMPLE	Perf(close, 20)

[TOP]

1.3.17.26 PerfD

PERFD
Average Daily Performance

Technical

SYNTAX	**PerfD(ARRAY array, ARRAY period)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the average daily performance of the array for the specified period (k-step Trend)
ADDITIONAL INFO	
EXAMPLE	PerfD(close, 20)

[TOP]

1.3.17.27 Rank

RANK
Rank

Technical

SYNTAX	**Rank(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the rank of the stock for this array, rank ranges are from 0 to 100 (To use in simulation)
ADDITIONAL INFO	
EXAMPLE	Rank(close)

[TOP]

1.3.17.28 Ref

REF
Lag

Technical

SYNTAX	**Ref(ARRAY array, ARRAY period)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	References a previous or subsequent element in a ARRAY. A negative period references X periods in the future; a positive period references X periods ago (Replace empty values in the ARRAY with NA)
ADDITIONAL INFO	
EXAMPLE	Ref(close, 1)

[TOP]

1.3.17.29 RefZero

REFZERO
Lag modified

Technical

SYNTAX	**RefZero(ARRAY array, ARRAY period)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	References a previous or subsequent element in a ARRAY. A negative period references X periods in the future; a positive period references X periods ago (Replace empty values in the ARRAY with 0)
ADDITIONAL INFO	
EXAMPLE	RefZero(close, 1)

[TOP]

1.3.17.30 RemoveSameSignals 0

REMOVESAMESIGNALS
Remove Excessive Signals

Technical

SYNTAX **RemoveSameSignals(ARRAY array, ARRAY array)**

RETURNS NUMERIC ARRAY

DESCRIPTION removes excessive signals.
Returns 1 on the first occurence of 'true' signal in the first array.
then Returns 0 until the second array is 'true' even if there are 'true' signals in the first array.

ADDITIONAL INFO

EXAMPLE RemoveSameSignals(array, array)

[TOP]

1.3.17.31 RemoveSameSignals 1

REMOVESAMESIGNALS
Remove Excessive Signals

Technical

SYNTAX	**RemoveSameSignals(ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	removes excessive signals. Returns 1 on the first occurence of 'true' signal in the first array. then Returns 0 until the second array become 'false' then 'true'.
ADDITIONAL INFO	
EXAMPLE	RemoveSameSignals(array)

[TOP]

1.3.17.32 RFun

RFUN
Relative Volatility

Technical

SYNTAX	**RFun(ARRAY array, ARRAY period)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the relative volatility of the specified array
ADDITIONAL INFO	
EXAMPLE	RFun(close, 30)

[TOP]

1.3.17.33 SumIf 0

SUMIF
Summation If

Technical

SYNTAX **SumIf(ARRAY condition, ARRAY array)**

RETURNS NUMERIC ARRAY

DESCRIPTION Calculates, when condition is TRUE, a cumulative sum of the ARRAY for all the lookback periods (current bar is included)

ADDITIONAL INFO

EXAMPLE SumIf(rsi(14) > 50, 1)

[TOP]

1.3.17.34 SumIf 1

SUMIF
Summation If

Technical

SYNTAX	**SumIf(ARRAY condition, ARRAY array, ARRAY period)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Calculates, when condition is TRUE, a cumulative sum of the ARRAY for the specified number of lookback periods (current bar is included)
ADDITIONAL INFO	
EXAMPLE	SumIf(rsi(14) > 50, 1, 20)

[TOP]

1.3.17.35 Value

VALUE
Value

Technical

SYNTAX	**Value(ARRAY array, ARRAY bar number)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the value of the specified ARRAY for the bar number X
ADDITIONAL INFO	
EXAMPLE	Value(close, 5)

[TOP]

1.3.17.36 ValueWhen 0

VALUEWHEN
Value When

Technical

SYNTAX	**ValueWhen(ARRAY expression, ARRAY array, ARRAY period)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the value of the array, when the expression is TRUE, of the n -th most recent occurrence
ADDITIONAL INFO	
EXAMPLE	ValueWhen(cross(close, ma(close, 20)), rsi(14), 10)

[TOP]

1.3.17.37 ValueWhen 1

VALUEWHEN
Value When

Technical

SYNTAX	**ValueWhen(ARRAY expression, ARRAY array)**
RETURNS	NUMERIC ARRAY
DESCRIPTION	Returns the value of the array, when the expression is TRUE, of the last occurrence
ADDITIONAL INFO	
EXAMPLE	ValueWhen(cross(close, ma(close, 20)), rsi(14))

[TOP]

1.4 QuantShare CSharp & JScript.Net

1.4.1 Advanced Money Management

1.4.1.1 OnEndPeriod

QuantShare API Documentation

OnEndPeriod Members

Manage your trading system using the advanced money management script.

Fields

TradingSystemSettings (of type **MMTradingSystemSettings**)	Trading system settings
Variables (of type **MMVariables**)	A class that is used to manipulate variables
Portfolio (of type **MMPortfolio**)	Metrics concerning the porftolio
Divers (of type **MMDivers**)	Divers functions
Data (of type **MMData**)	A class that is used to retrieve data
Orders (of type **MMOrders**)	A class that is used to generate orders
Functions (of type **MMOnEndPeriod**)	Functions related to the OnEndPeriod event

[TOP]

1.4.1.2 OnEndSimulation

QuantShare API Documentation

OnEndSimulation Members

Manage your trading system using the advanced money management script.

Fields

TradingSystemSettings (of type **MMTradingSystemSettings**)	Trading system settings
Variables (of type **MMVariables**)	A class that is used to manipulate variables
Portfolio (of type **MMPortfolio**)	Metrics concerning the porftolio
Divers (of type **MMDivers**)	Divers functions
Functions (of type **MMOnEndSimulation**)	Functions related to the OnEndSimulation event

[TOP]

1.4.1.3 OnStartSimulation

QuantShare API Documentation

OnStartSimulation Members

Manage your trading system using the advanced money management script.

Fields

TradingSystemSettings (of type **MMTradingSystemSettings**)	Trading system settings
Variables (of type **MMVariables**)	A class that is used to manipulate variables
Portfolio (of type **MMPortfolio**)	Metrics concerning the porftolio
Divers (of type **MMDivers**)	Divers functions
Functions (of type **MMOnStartSimulation**)	Functions related to the OnStartSimulation event
Optimize (of type **MMOptimize**)	A class that can be used to define variables that will be used in the optimization

[TOP]

1.4.1.4 OnClosePosition

QuantShare API Documentation

OnClosePosition Members

Manage your trading system using the advanced money management script.

Fields

TradingSystemSettings (of type **MMTradingSystemSettings**)	Trading system settings
Variables (of type **MMVariables**)	A class that is used to manipulate variables
Portfolio (of type **MMPortfolio**)	Metrics concerning the porftolio
Divers (of type **MMDivers**)	Divers functions
Data (of type **MMData**)	A class that is used to retrieve data
Orders (of type **MMOrders**)	A class that is used to generate orders
Functions (of type **MMOnClosePosition**)	Functions related to the OnClosePosition event

[TOP]

1.4.1.5 OnNewPosition

QuantShare API Documentation

OnNewPosition Members

Manage your trading system using the advanced money management script.

Fields

TradingSystemSettings (of type **MMTradingSystemSettings**)	Trading system settings
Variables (of type **MMVariables**)	A class that is used to manipulate variables
Portfolio (of type **MMPortfolio**)	Metrics concerning the porftolio
Divers (of type **MMDivers**)	Divers functions
Data (of type **MMData**)	A class that is used to retrieve data
Orders (of type **MMOrders**)	A class that is used to generate orders
NewPosition (of type **MMNewPosition**)	Class containing information about the position the simulator is about to buy/short
Functions (of type **MMOnNewPosition**)	Functions related to the OnNewPosition event

[TOP]

1.4.1.6 Members

1.4.1.6.1 MMDivers

QuantShare API Documentation

MMDivers Members

Divers functions

Parent

Properties

IsFirstBar (of type **Boolean**)	Specify whether the current bar is the first one or not
CurrentDate (of type **DateTime**)	Current Date

Methods

Output (returns **Void**)	(parameterless) Set Output
MovingAverage (returns **Double**)	(array **Double[]**, period **Int32**) Calculate the moving average of the provided array and for the specified period
Sum (returns **Double**)	(array **Double[]**, period **Int32**) Calculate the sum of the provided array and for the specified period
Max (returns **Double**)	(array **Double[]**, period **Int32**) Calculate the maximum value of the provided array and for the specified period
Min (returns **Double**)	(array **Double[]**, period **Int32**) Calculate the minimum value of the provided array and for the specified period

[TOP]

1.4.1.6.2 MMOnEndPeriod

QuantShare API Documentation

MMOnEndPeriod Members

Functions related to the OnEndPeriod event

Parent

Properties

AvailableCashForLongPositions (of type **Double**)	Get available cash for long positions
AvailableCashForShortPositions (of type **Double**)	Get available cash for short positions
NbAvailableLongPositions (of type **Int32**)	Get available number of long positions
NbAvailableShortPositions (of type **Int32**)	Get available number of short positions

Methods

AddMetric (returns **Void**)	(Metric name **String**, Metric value for the current period (date) **Double**) Add a time-series metric to the simulation
UpdateNumberOfPositions (returns **Void**)	(number of positions **Int32**) Update the maximum number of positions within the portfolio
UpdateMarginFactor (returns **Int32**)	(margin factor **Double**) Update the margin factor
UpdateMarginFactor (returns **Int32**)	(margin factor (higher or equal to 1) **Double**, order type used to scale-in or scale-out positions **_TradingOrder**, do not scale positions if the position size is less than minPerPosition **Double**) Update the margin factor and scale-in or scale-out existing positions to meet the new requirements
UpdatePercentInvested (returns **Int32**)	(percent invested (between 0 and 100) **Double**) Update the percentage invested value
UpdatePercentInvested (returns **Int32**)	(percent invested (between 0 and 100) **Double**, order type used to scale-in or scale-out positions **_TradingOrder**, do not scale positions if the position size is less than minPerPosition **Double**) Update the percentage invested value and scale-in or scale-out existing positions to meet the new requirements
SkipPeriods (returns **Void**)	(Number of periods **Int32**) Skip money monagement execution for the specified number of periods
RejectNewPositionsDuringTheNextPeriods (returns **Void**)	(Number of periods **Int32**)

	Instruct the simulator to not open new positions during the specified number of periods
CloseAllPositions (returns **Void**)	(Number of periods to skip executing OnEndPeriod script **Int32**) Close all portfolio's positions
CloseAllPositions (returns **Void**)	(Number of periods to skip executing OnEndPeriod script **Int32**, Trading order **_TradingOrder**) Close all portfolio's positions
GetNumberShares (returns **Int32**)	(Symbol name **String**, Long or Short **Boolean**, Trading order **_TradingOrder**) Get the approximative number of shares to be bought or shorted
AddLongPosition (returns **Boolean**)	(Symbol name **String**, Number of shares **Int32**, Custom value **Double**, Trading order **_TradingOrder**) Add a new long position
AddLongPosition (returns **Boolean**)	(Symbol name **String**, Custom value **Double**, Trading order **_TradingOrder**) Add a new long position - Let the simulator automatically choose the number of shares to buy
AddShortPosition (returns **Boolean**)	(Symbol name **String**, Number of shares **Int32**, Custom value **Double**, Trading order **_TradingOrder**) Add a new short position
AddShortPosition (returns **Boolean**)	(Symbol name **String**, Custom value **Double**, Trading order **_TradingOrder**) Add a new short position - Let the simulator automatically choose the number of shares to short
SellPosition (returns **Boolean**)	(Symbol name **String**, Number of shares **Int32**, Custom value **Double**, Trading order **_TradingOrder**) Sell a long position
SellPosition (returns **Boolean**)	(Symbol name **String**, Custom value **Double**, Trading order **_TradingOrder**) Sell a long position - Sell all shares
CoverPosition (returns **Boolean**)	(Symbol name **String**, Number of shares **Int32**, Custom value **Double**, Trading order **_TradingOrder**) Cover a short position
CoverPosition (returns **Boolean**)	(Symbol name **String**, Custom value **Double**, Trading order **_TradingOrder**) Cover a short position - Cover all shares

[TOP]

1.4.1.6.3 MMOnEndSimulation

QuantShare API Documentation

MMOnEndSimulation Members

Functions related to the OnEndSimulation event

Parent

Methods

AddMetric (returns **Void**)	(Metric name **String**, Metric value **Double**) Add a metric to the simulator

[TOP]

1.4.1.6.4 MMOptimize

QuantShare API Documentation

MMOptimize Members

A class that can be used to define variables that will be used in the optimization

Parent

Methods

OptimizeText (returns **Void**)	(variable name **String**, values **String[]**) Create an optimizable variable, use the Variable Class to get the value
OptimizeDouble (returns **Void**)	(variable name **String**, start value **Double**, end value **Double**, step value **Double**) Create an optimizable variable, use the Variable Class to get the value

[TOP]

1.4.1.6.5 MMPortfolio

QuantShare API Documentation

MMPortfolio Members

Metrics concerning the porftolio

Parent

Properties

Equity (of type **Double**)	Current portfolio equity
Drawdown (of type **Double**)	Current portfolio drawdown
PercentageInvested (of type **Double**)	Current portfolio percentage of invested money
NbPositions (of type **Int32**)	Current number of positions within the portfolio

Methods

IsInPortfolio (returns **Boolean**)	(symbol name **String**, Long or Short position **Boolean**) Returns whether a security symbol is in the portfolio
IsInPendingOrders (returns **Boolean**)	(symbol name **String**, Long or Short list **Boolean**) Returns whether a security symbol is in the pending orders list
GetPosition (returns **MMPosition**)	(symbol name **String**) Get a position from the portfolio
GetAllPositions (returns **MMPosition[]**)	(parameterless) Get all positions generated by the simulator - open and closed positions
GetLastClosedPositions (returns **MMPosition[]**)	(parameterless) Get the positions that were closed in the last period
GetOpenPositions (returns **MMPosition[]**)	(parameterless) Get the current portfolio positions

[TOP]

1.4.1.6.6 MMTradingSystemSettings

QuantShare API Documentation

MMTradingSystemSettings Members

Trading system settings

Parent

Properties

InitialEquity (of type **Double**)	Portfolio initial equity
StartDate (of type **DateTime**)	Simulation start date
EndDate (of type **DateTime**)	Simulation end date
NbPositions (of type **Int32**)	Portfolio maximum number of positions
PercentToInvest (of type **Double**)	Portfolio percent of capital to invest

[TOP]

1.4.1.6.7 MMVariables

QuantShare API Documentation

MMVariables Members

A class that is used to manipulate variables

Parent

Properties

[String] (of type **Object**)	Get a variable value

Methods

SetVariable (returns **Void**)	(Variable Name **String**, Value **Object**) Assign a value to a variable
GetVariable (returns **Object**)	(Variable Name **String**) Get a variable value
IsVariableExists (returns **Boolean**)	(Variable Name **String**) Returns true if the variable exist, otherwise returns false

[TOP]

1.4.1.6.8 MMOnNewPosition

QuantShare API Documentation

MMOnNewPosition Members

Functions related to the OnNewPosition event

Parent

Methods

GetNextTrade (returns **Void**)	(parameterless) Reject the current trade
RejectAllTradesForThisPeriod (returns **Void**)	(parameterless) Reject all trades for the current period (date)
UpdateNumberOfShares (returns **Boolean**)	(Number of shares **Int32**) Update the number of shares to buy/short

[TOP]

1.4.1.6.9 MMExitPosition

QuantShare API Documentation

MMExitPosition Members

Parent

Properties

EntryDate (of type **DateTime**)	Current position entry date
SignalEntryDate (of type **DateTime**)	Current position signal date. Signal date is the date where the symbol buy/short value became TRUE. Entry date is the date where the symbol was bought or shorted. The Signal date could be different than the Entry date when the buy/short entry price could not be met on the signal date.
Symbol (of type **String**)	Current position symbol
PositionEquity (of type **Double**)	Current position size
Performance (of type **Double**)	Current position performance
EntryPrice (of type **Double**)	Current position entry price
BarsSinceEntry (of type **Int32**)	Current position number of bars held
MAE (of type **Double**)	Current position maximum adverse excursion
MFE (of type **Double**)	Current position maximum favorable excursion
NbShares (of type **Int32**)	Current position number of shares
MaximumDrawdown (of type **Double**)	Current position maximum drawdown
NumberOfScaleIn (of type **Int32**)	Current position number of scale-in trades
NumberOfScaleOut (of type **Int32**)	Current position number of scale-out trades
LastPrice (of type **Double**)	Current position last price
Var1 (of type **Double**)	Custom variable
Var2 (of type **Double**)	Custom variable
Var3 (of type **Double**)	Custom variable
Var4 (of type **Double**)	Custom variable

Methods

GetValueFromFormula (returns **Double**)	(parameterless) Parse a vector-based formula and returns the vector value for the current date. Example: GetValueFromFormula('a = rsi(14);', 'a'); -> to get the RSI value of the current symbol for the current date.

[TOP]

1.4.1.6.10 MMNewPosition

QuantShare API Documentation

MMNewPosition Members

Class containing information about the position the simulator is about to buy/short

Parent

Properties

IsLong (of type **Boolean**)	Specify whether the new position is Long or Short
Price (of type **Double**)	The new position price
NbShares (of type **Int32**)	The number of shares for the current position
RemainingCash (of type **Double**)	The remaining portfolio cash before taking this position
NbPositions (of type **Int32**)	The number of positions in the portfolio

[TOP]

1.4.1.6.11 MMOnClosePosition

QuantShare API Documentation

MMOnClosePosition Members

Functions related to the OnClosePosition event

Parent

Methods

AddTradeMetric (returns **Void**)	(Metric name **String**, Metric value **Double**) Add a trade metric
GetPositionDetails (returns **MMExitPosition**)	(parameterless) Get details about the current position
DontCloseTheCurrentPosition (returns **Void**)	(parameterless) Restrict the simulator from exiting the current position
DontClosePositionsForThisPeriod (returns **Void**)	(parameterless) Restrict the simulator from exiting all the positions within this period (date)
UpdateNumberOfShares (returns **Boolean**)	(Number of shares **Int32**) Update the number of shares to sell/cover

[TOP]

1.4.1.6.12 MMPosition

QuantShare API Documentation

MMPosition Members

Parent

Properties

EntryDate (of type **DateTime**)	Current position entry date
SignalEntryDate (of type **DateTime**)	Current position signal date. Signal date is the date where the symbol buy/short value became TRUE. Entry date is the date where the symbol was bought or shorted. The Signal date could be different than the Entry date when the buy/short entry price could not be met on the signal date.
Symbol (of type **String**)	Current position symbol
PositionEquity (of type **Double**)	Current position size
Performance (of type **Double**)	Current position performance
EntryPrice (of type **Double**)	Current position entry price
BarsSinceEntry (of type **Int32**)	Current position number of bars held
MAE (of type **Double**)	Current position maximum adverse excursion
MFE (of type **Double**)	Current position maximum favorable excursion
NbShares (of type **Int32**)	Current position number of shares
MaximumDrawdown (of type **Double**)	Current position maximum drawdown
NumberOfScaleIn (of type **Int32**)	Current position number of scale-in trades
NumberOfScaleOut (of type **Int32**)	Current position number of scale-out trades
LastPrice (of type **Double**)	Current position last price
Var1 (of type **Double**)	Custom variable
Var2 (of type **Double**)	Custom variable
Var3 (of type **Double**)	Custom variable
Var4 (of type **Double**)	Custom variable

Methods

ScaleIn (returns **Void**)	(Number of shares **Int32**) Scale-in position
ScaleIn (returns **Void**)	(Number of shares **Int32**, Trading order **_TradingOrder**) Scale-in position
ScaleOut (returns **Void**)	(Number of shares **Int32**) Scale-out position
ScaleOut (returns **Void**)	(Number of shares **Int32**, Trading order **_TradingOrder**) Scale-out position
ClosePosition (returns **Void**)	(parameterless) Close the current position
ClosePosition (returns **Void**)	(Open market order **_OpenMarketOrder**) Close the current position
ClosePosition (returns **Void**)	(Close market order **_CloseMarketOrder**) Close the current position
ClosePosition (returns **Void**)	(Limit order **_LimitOrder**) Close the current position
GetValueFromFormula (returns **Double**)	(parameterless) Parse a vector-based formula and returns the vector value for the current date. Example: GetValueFromFormula('a = rsi(14);', 'a'); -> to get the RSI value of the current symbol for the current date.

[TOP]

1.4.1.6.13 TimeSeries

QuantShare API Documentation

TimeSeries Members

Parent

Properties

Count (of type **Int32**)	Get the time-series number of elements
[Int32] (of type **Double**)	Get a time-series value

Methods

GetValue (returns **Double**)	(lag **Int32**) Get a time-series value

[TOP]

1.4.1.6.14 VectorCustom

QuantShare API Documentation

VectorCustom Members

Parent

Properties

Count (of type **Int32**)	Get the number of elements
[Int32] (of type **CustomDatabaseValues**)	Get the elements at the specified bar index

Methods

GetValue (returns **CustomDatabaseValues**)	(lag **Int32**) Get the elements at the specified bar index

[TOP]

1.4.1.6.15 CustomDatabaseValues

QuantShare API Documentation

CustomDatabaseValues Members

Parent

Properties

[Int32] (of type **Object**)	Get the element at the specified index
Length (of type **Int32**)	Get the number of elements for the current bar period

Methods

GetValue (returns **Object**)	(index **Int32**) Get the element at the specified index

[TOP]

1.4.1.6.16 MMData

QuantShare API Documentation

MMData Members

A class that is used to retrieve data

Parent

Methods

GetCustomDatabaseData (returns **VectorCustom**)	(database **String**, field **String**, symbol name **String**) Get the specified database field data
ParseFormula (returns **MMParser**)	(formula. Ex: a = rsi(14); **String**) Parse a vector-based formula
IsMarketData (returns **Boolean**)	(Symbol name **String**) Returns whether there is market data or not on the current date
GetPriceSeries (returns **TimeSeries**)	(parameterless) Returns a price series
IsSymbolExists (returns **Boolean**)	(symbol name **String**) Returns TRUE if the specified symbol exists
GetSymbols (returns **String[]**)	(parameterless) Get the strategy's symbols
GetBuySignals (returns **MMEntrySignals**)	(parameterless) Get buy signals generated at the end of this period
GetShortSignals (returns **MMEntrySignals**)	(parameterless) Get short signals generated at the end of this period
GetSellSignals (returns **MMExitSignals**)	(parameterless) Get sell signals generated at the end of this period
GetCoverSignals (returns **MMExitSignals**)	(parameterless) Get cover signals generated at the end of this period

1.4.1.6.17 MMOrders

QuantShare API Documentation

MMOrders Members

A class that is used to generate orders

Parent

Methods

OpenMarketOrder (returns **_TradingOrder**)	(parameterless) Create an market order that is executed at the open
OpenMarketOrder (returns **_TradingOrder**)	(execute the order after the specified number of bars **Int32**, keep the order active for the specified number of bars **Int32**) Create an market order that is executed at the open
CloseMarketOrder (returns **_TradingOrder**)	(parameterless) Create an market order that is executed at the close
CloseMarketOrder (returns **_TradingOrder**)	(execute the order after the specified number of bars **Int32**, keep the order active for the specified number of bars **Int32**) Create an market order that is executed at the close
LimitOrder (returns **_TradingOrder**)	(limit price **Double**) Create an market order that is executed at the specified price
LimitOrder (returns **_TradingOrder**)	(limit price **Double**, execute the order after the specified number of bars **Int32**, keep the order active for the specified number of bars **Int32**) Create an market order that is executed at the specified price

[TOP]

1.4.1.6.18 MMOnStartSimulation

QuantShare API Documentation

MMOnStartSimulation Members

Functions related to the OnStartSimulation event

Parent

Methods

SetTextInput (returns **Void**)	(Variable name **String**, Default Value **String**, Description **String**) Get a text value for the Money Management Input Form, use the Variable Class to get the value
SetTextInput (returns **Void**)	(Variable name **String**, Default Value **String**, Description **String**, Set the possible values **String[]**) Get a text value for the Money Management Input Form, use the Variable Class to get the value
SetNumericInput (returns **Void**)	(Variable name **String**, Default Value **Double**, Description **String**) Get a numeric value for the Money Management Input Form, use the Variable Class to get the value

[TOP]

1.4.1.6.19 MMParser

QuantShare API Documentation

MMParser Members

Parent

Properties

IsErrors (of type **Boolean**)	Returns whether an error occured while parsing the formula

Methods

GetErrors (returns **String**)	(parameterless) Get the last execution errors
GetTimeSeries (returns **TimeSeries**)	(symbol name **String**, variable within the formula **String**) Retrieves a Time-Series.

[TOP]

1.4.1.6.20 MMEntrySignals

QuantShare API Documentation

MMEntrySignals Members

Parent

Properties

Count (of type **Int32**)	Get the number of signals

Methods

GetSymbol (returns **String**)	(Signal index **Int32**) Get the symbol name
GetDetail (returns **Double**)	(Signal index **Int32**, Tags: price, rank, stoploss, stoptrailing, stopprofit, stopbars **String**) Get a symbol detail

[TOP]

1.4.1.6.21 MMExitSignals

QuantShare API Documentation

MMExitSignals Members

Parent

Properties

Count (of type **Int32**)	Get the number of signals

Methods

GetSymbol (returns **String**)	(Signal index **Int32**) Get the symbol name
GetPrice (returns **Double**)	(Signal index **Int32**) Get the symbol sell/cover price

1.4.1.6.22 _TradingOrder

QuantShare API Documentation

_TradingOrder Members

Parent

Properties

OrderLag (of type **Int32**)	Execute order after the number of bars specified in the OrderLag variable
PositionValidFor (of type **Int32**)	For entry orders: The number of periods to keep the order alive (in case the order isn't filled) For exit orders: The number of periods to keep the order alive before transforming it to an open market order

[TOP]

1.4.2 Analyze Ranking System Metrics

QuantShare API Documentation

Analyze Ranking System Metrics Members

Fields

◆ Buckets (of type **RankingSystemPerformanceDetails**)	Class containing information about the ranking system buckets
◆ Fitness (of type **Double**)	Fitness value

[TOP]

1.4.2.1 Analyze Ranking System Metrics

Analyze Ranking System Metrics Members

Fields

Buckets (of type **RankingSystemPerformanceDetails**)	Class containing information about the ranking system buckets
Fitness (of type **Double**)	Fitness value

[TOP]

1.4.2.2 Members

1.4.2.2.1 BucketDetails

QuantShare API Documentation

BucketDetails Members

Parent

Properties

AnnualReturn (of type **Double**)	Annual return
NbPositions (of type **Int32**)	Number of positions taken within this bucket
Label (of type **String**)	Bucket label
Score (of type **Double**)	Bucket score
DetailsDate (of type **BucketDetailsDate[]**)	Bucket's positions details for rebalance dates. Each array element corresponds to a rebalance date.
RebalancePeriod (of type **Int32**)	The ranking system rebalance period

[TOP]

1.4.2.2.2 BucketDetailsDate

QuantShare API Documentation

BucketDetailsDate Members

Parent

Properties

AvgPerf (of type **Double**)	Average performance
AvgBarsHeld (of type **Double**)	Average bars held
Equity (of type **Double**)	Equity value
NbPositions (of type **Int32**)	Number of positions
Date (of type **DateTime**)	Rebalance date
Score (of type **Double**)	Score value for the current bucket and the current rebalance date

[TOP]

1.4.2.2.3 RankingSystemPerformanceDetails

QuantShare API Documentation

RankingSystemPerformanceDetails Members

Class containing information about the ranking system buckets

Parent

Properties

Buckets (of type **BucketDetails[]**)	Array containing the buckets details, the first array element contains the data of the first bucket...
LastBucket (of type **BucketDetails**)	Last bucket details

[TOP]

1.4.3 Analyze Rules Metrics

QuantShare API Documentation

Analyze Rules Metrics Members

Fields

Output (of type **Double**)	Average output per position
OutputPerBar (of type **Double**)	Average output per position and per bar
PercentPositive (of type **Double**)	Percent of positive positions
BestPosition (of type **Double**)	Best position output
WorstPosition (of type **Double**)	Worst position output
AvgBarsHeld (of type **Double**)	Average bars held per position
NbPositions (of type **Double**)	Number of positions
StandardDeviation (of type **Double**)	Standard deviation of the position outputs
Positions (of type **SymbolPositionDetails**)	Details of all positions
Fitness (of type **Double**)	Fitness value
Functions (of type **RulesFunctions**)	Function class that lets you add additional metrics

[TOP]

1.4.3.1 Analyze Rules Metrics

QuantShare API Documentation

Analyze Rules Metrics Members

Fields

Output (of type **Double**)	Average output per position
OutputPerBar (of type **Double**)	Average output per position and per bar
PercentPositive (of type **Double**)	Percent of positive positions
BestPosition (of type **Double**)	Best position output
WorstPosition (of type **Double**)	Worst position output
AvgBarsHeld (of type **Double**)	Average bars held per position
NbPositions (of type **Double**)	Number of positions
StandardDeviation (of type **Double**)	Standard deviation of the position outputs
Positions (of type **SymbolPositionDetails**)	Details of all positions
Fitness (of type **Double**)	Fitness value
Functions (of type **RulesFunctions**)	Function class that lets you add additional metrics

[TOP]

1.4.3.2 Members

1.4.3.2.1 PositionDetails

QuantShare API Documentation

PositionDetails Members

Parent

Properties

Output (of type **Double**)	The position output
NbBars (of type **Int32**)	The number of bars the current position was held
Date (of type **DateTime**)	Position entry date

[TOP]

1.4.3.2.2 SymbolPositionDetails

QuantShare API Documentation

SymbolPositionDetails Members

Details of all positions

Parent

Properties

Symbol (of type **String**)	Symbol name
Positions (of type **PositionDetails[]**)	Positions details for the current symbol

[TOP]

1.4.3.2.3 RulesFunctions

QuantShare API Documentation

RulesFunctions Members

Function class that lets you add additional metrics

Parent

Methods

AddMetric (returns **Void**)	(metric name **String**, metric value **Double**) Add a new metric
GetOutputName (returns **String**)	(parameterless) Returns the output name
GetRuleFormula (returns **String**)	(parameterless) Returns the rule formula

[TOP]

1.4.4 Parser Post-Script

QuantShare API Documentation

Parser Post-Script Members

Script that is executed after parsing the downloaded data.

Fields

Data (of type **PostScript**)	Contains the data that was parsed

[TOP]

1.4.4.1 Downloader Post-Script

QuantShare API Documentation

Parser Post-Script Members

Script that is executed after parsing the downloaded data.

Fields

Data (of type **PostScript**)	Contains the data that was parsed

[TOP]

1.4.4.2 Members

1.4.4.2.1 PostScript

QuantShare API Documentation

PostScript Members

Contains the data that was parsed

Parent

Properties

Rows (of type **PostScriptRow[]**)	Get the content rows

[TOP]

1.4.4.2.2 PostScriptRow

QuantShare API Documentation

PostScriptRow Members

Parent

Properties

IsIgnoreLine (of type **Boolean**)	Specify whether to ignore the current row or not
Data (of type **Object[]**)	Get the elements of the current row

[TOP]

1.4.5 Parser Pre-Script

QuantShare API Documentation

Parser Pre-Script Members

Script that is executed before parsing the downloaded data.

Fields

Content (of type **PreScript**)	Contains the content to parse

[TOP]

1.4.5.1 Downloader Pre-Script

QuantShare API Documentation

Parser Pre-Script Members

Script that is executed before parsing the downloaded data.

Fields

Content (of type **PreScript**)	Contains the content to parse

[TOP]

1.4.5.2 Members

1.4.5.2.1 PreScript

QuantShare API Documentation

PreScript Members

Contains the content to parse

Parent

Properties

Rows (of type **PreScriptRow[]**)	Get the content rows

Methods

GetURLORFileName (returns **String**)	(parameterless) Get the URL or the file path that was used to get the current content
GetSymbolNameFromNameX (returns **String**)	(0 for name1... **Int32**, symbol nameX **String**) Get symbol name from name1, name2 or name3
AddRow (returns **Void**)	(Elements to add **String[]**) Add a new row

[TOP]

1.4.5.2.2 PreScriptRow

QuantShare API Documentation

PreScriptRow Members

Parent

Properties

IsIgnoreLine (of type **Boolean**)	Specify whether to ignore the current line or not
Data (of type **String[]**)	Get the elements of the current line

[TOP]

1.4.6 Downloader URL-Script

QuantShare API Documentation

Downloader URL-Script Members

This script is used to generated custom URLs. These URLs will be used to download data.

Fields

Functions (of type **URLScriptFunctions**)	Contains a list of methods

[TOP]

1.4.6.1 Downloader URL-Script

QuantShare API Documentation

Downloader URL-Script Members

This script is used to generated custom URLs. These URLs will be used to download data.

Fields

Functions (of type **URLScriptFunctions**)	Contains a list of methods

[TOP]

1.4.6.2 Members

1.4.6.2.1 Net

QuantShare API Documentation

Net Members

Parent

Methods

DownloadString (returns **String**)	(URL **String**) Download content from the specified URL
GetLinks (returns **String[]**)	(html content **String**) Get all links from the provided content
GetLinksByAnchor (returns **String[]**)	(html content **String**, anchor text **String**) Get links from the specified content that contains the provided anchor text
GetHTMLElements (returns **HTMLElement[]**)	(parameterless) Get HTML elements by tag name
GetHTMLElementsByTag (returns **HTMLElement[]**)	(html content **String**, anchor text **String**) Get HTML elements by tag name

[TOP]

1.4.6.2.2 URLScriptFunctions

QuantShare API Documentation

URLScriptFunctions Members

Contains a list of methods

Parent

Properties

Net (of type **Net**)	Functions to download, parse... HTML content

Methods

GetDefaultURL (returns **String**)	(parameterless) Returns the default URL
GetStartDate (returns **DateTime**)	(parameterless) Get the start date
GetEndDate (returns **DateTime**)	(parameterless) Get the end date
CreateURLPath (returns **URLPath**)	(parameterless) Create an URL Path class, the downloader must load all the URLs specified in this class in order to download your content
AddURL (returns **Void**)	(URL **String**, Message to display **String**) Add an URL to the download list
AddURLPath (returns **Void**)	(URL Path **URLPath**) Add an URL Path, the downloader must load all the specified URLs in order to download your content
GetValues (returns **String**)	(Parameter name **String**) Get an URL parameter value. Example: The current symbol
SetValues (returns **Void**)	(Parameter name **String**, Parameter value **String**) Set an URL parameter value. Example: The current symbol
GetAllValues (returns **String[]**)	(Parameter name **String**) Get all the URL parameter values. Example: All the selected symbols

[TOP]

1.4.6.2.3 HTMLElement

QuantShare API Documentation

HTMLElement Members

Parent

Properties

TagName (of type **String**)	Get the tag name
InnerText (of type **String**)	Get the inner text
InnerHTML (of type **String**)	Get the inner html

Methods

GetAttribute (returns **String**)	(attribute name **String**) Get an attribute value

1.4.6.2.4 URLPath

QuantShare API Documentation

URLPath Members

Parent

Methods

AddURL (returns **Void**)	(URL **String**, Message to display **String**) Add an URL to the current Path
AddURL (returns **Void**)	(URL **String**, Message to display **String**, Referer URL **String**) Add an URL to the current Path

[TOP]

1.4.7 Indicators

Indicators Members

Create an indicator using JScript.Net.

Fields

◆ result (of type **VectorD**)	You have to assign values to this array
◆ cFunctions (of type **CFunctions**)	Divers functions

[TOP]

1.4.7.1 Indicators

QuantShare API Documentation

Indicators Members

Create an indicator using JScript.Net.

Fields

result (of type **VectorD**)	You have to assign values to this array
cFunctions (of type **CFunctions**)	Divers functions

[TOP]

1.4.7.2 Members

1.4.7.2.1 CFunctions

CFunctions Members

Divers functions

Parent

Properties

Close (of type **VectorD**)	Close price vector
Date (of type **VectorDate**)	Date vector
High (of type **VectorD**)	High price vector
Low (of type **VectorD**)	Low price vector
Open (of type **VectorD**)	Open price vector
Volume (of type **VectorD**)	Volume price vector
Symbol (of type **String**)	Symbol name
OpenInt (of type **VectorD**)	Open Interest price vector

Methods

SetForwardAndBackwardBars (returns **Void**)	(backward bars **Int32**, forward bars **Int32**) Set the number of forward and backward bars used by this function□Theses values are used by the application to optimize the parsing process Example: sma(close, 10), uses 10 backward bars and 0 forward bars
CreateNumericVector (returns **VectorD**)	(parameterless) Returns an empty numeric vector
CreateTextVector (returns **VectorS**)	(parameterless) Returns an empty text vector
GetCustomDatabaseData (returns **VectorCustom**)	(database **String**, field **String**) For each symbol bar get the elements of the database field data that occured during this bar interval
GetCustomDatabaseData (returns **VectorCustom**)	(database **String**, field **String**, symbol name **String**) For each symbol bar get the elements of the database field data that occured during this bar interval

CompileFormula (returns **QSFormula**)	(formula **String**) Compile a formula
GetVectorDouble (returns **VectorD**)	(variable name **String**) Get a numeric vector from the current formula
GetVectorString (returns **VectorS**)	(variable name **String**) Get a text vector from the current formula

[TOP]

1.4.7.2.2 CustomDatabaseValues

QuantShare API Documentation

CustomDatabaseValues Members

Parent

Properties

[Int32] (of type **Object**)	Get the element at the specified index
Length (of type **Int32**)	Get the number of elements for the current bar period

Methods

GetValue (returns **Object**)	(index **Int32**) Get the element at the specified index

[TOP]

1.4.7.2.3 QSFormula

QuantShare API Documentation

QSFormula Members

Parent

Methods

GetVectorDouble (returns **VectorD**)	(variable name **String**) Get a double variable from the compiled formula
GetVectorString (returns **VectorS**)	(variable name **String**) Get a string variable from the compiled formula

[TOP]

1.4.7.2.4 VectorCustom

QuantShare API Documentation

VectorCustom Members

Parent

Properties

[Int32] (of type **CustomDatabaseValues**)	Get the elements at the specified bar index
Length (of type **Int32**)	Get the vector length

Methods

GetValues (returns **CustomDatabaseValues**)	(index **Int32**) Get the elements at the specified bar index

[TOP]

1.4.7.2.5 VectorD

QuantShare API Documentation

VectorD Members

Parent

Properties

Length (of type **Int32**)	Get the vector length

Methods

SetValue (returns **Void**)	(index **Int32**, value **Double**) Assign a value to an element of the vector
GetValue (returns **Double**)	(index **Int32**) Get an element value from the vector
Assign (returns **Void**)	(value **Double**) Assign a value to all the elements of the vector

[TOP]

1.4.7.2.6 VectorDate

QuantShare API Documentation

VectorDate Members

Parent

Properties

Length (of type **Int32**)	Get the vector length

Methods

SetValue (returns **Void**)	(index **Int32**, value **DateTime**) Assign a value to an element of the vector
GetValue (returns **DateTime**)	(index **Int32**) Get an element value from the vector

[TOP]

1.4.7.2.7 VectorS

QuantShare API Documentation

VectorS Members

Parent

Properties

Length (of type **Int32**)	Get the vector length

Methods

SetValue (returns **Void**)	(index **Int32**, value **String**) Assign a value to an element of the vector
GetValue (returns **String**)	(index **Int32**) Get an element value from the vector
Assign (returns **Void**)	(value **String**) Assign a value to all the elements of the vector

[TOP]

1.4.8 Perdiction Model Metrics

QuantShare API Documentation

Perdiction Model Metrics Members

Fields

Cycle (of type **Int32**)	The current cycle number
NetworkError (of type **Double**)	The network error
LearningErrors (of type **PredictionErrors**)	Metrics for the learning samples
ValidationErrors (of type **PredictionErrors**)	Metrics for the validation samples
TestErrors (of type **PredictionErrors**)	Metrics for the test samples
Fitness (of type **Double**)	The fitness value

[TOP]

1.4.8.1 Perdiction Model Metrics

QuantShare API Documentation

Perdiction Model Metrics Members

Fields

Cycle (of type **Int32**)	The current cycle number
NetworkError (of type **Double**)	The network error
LearningErrors (of type **PredictionErrors**)	Metrics for the learning samples
ValidationErrors (of type **PredictionErrors**)	Metrics for the validation samples
TestErrors (of type **PredictionErrors**)	Metrics for the test samples
Fitness (of type **Double**)	The fitness value

[TOP]

1.4.8.2 Members

1.4.8.2.1 PredictionErrors

QuantShare API Documentation

PredictionErrors Members

Metrics for the test samples

Parent

Properties

MSE (of type **Double**)	Mean squared error
POCD (of type **Double**)	Prediction of correct direction
POCID (of type **Double**)	Prediction of change in direction
UTHEIL (of type **Double**)	Theil's U statistic assesses predictive accuracy relative to a naive no-change model. It is unitary when the MSFE (mean absolute forecast error) equals the mean square error of naive no-change forecasts, and it is greater than 1.0 if predictions are less accurate than no-change forecasts.

[TOP]

1.4.9 Simulation Metrics

QuantShare API Documentation

Simulation Metrics Members

Fields

InitialCapital (of type **Double**)	Initial Capital
EndCapital (of type **Double**)	End Capital
NetProfit (of type **Double**)	Net Profit
NetProfitInPercentage (of type **Double**)	Net Profit In Percentage
Exposure (of type **Double**)	Exposure
NetRiskAdjustedReturn (of type **Double**)	Net Risk Adjusted Return
AnnualReturn (of type **Double**)	Annual Return
RiskAdjustedReturn (of type **Double**)	Risk Adjusted Return
AverageProfitLoss (of type **Double**)	Average ProfitLoss
AverageProfitLossInPercentage (of type **Double**)	Average ProfitLoss In Percentage
AverageBarsHeld (of type **Double**)	Average Bars Held
TotalProfit (of type **Double**)	Total Profit
NumberOfWinners (of type **Double**)	Number Of Winners
AverageProfit (of type **Double**)	Average Profit
AverageProfitInPercentage (of type **Double**)	Average Profit In Percentage
AverageBarsHeldForWinners (of type **Double**)	Average Bars Held For Winners
MaxConsecutiveWinner (of type **Double**)	Max Consecutive Winner
LargestWinner (of type **Double**)	Largest Winner
NumberOfBarsInLargestWinner (of type **Double**)	Number Of Bars In Largest Winner
TotalLoss (of type **Double**)	Total Loss

NumberOfLosers (of type **Double**)	Number Of Losers
AverageLoss (of type **Double**)	Average Loss
AverageLossInPercentage (of type **Double**)	Average Loss In Percentage
AverageBarsHeldForLosers (of type **Double**)	Average Bars Held For Losers
MaxConsecutiveLoser (of type **Double**)	Max Consecutive Loser
LargestLoser (of type **Double**)	Largest Loser
NumberOfBarsInLargestLoser (of type **Double**)	Number Of Bars In Largest Loser
MaximumTradeDrawdown (of type **Double**)	Maximum Trade Drawdown
MaximumTradeDrawdownInpercentage (of type **Double**)	Maximum Trade Drawdown In percentage
MaximumSystemDrawdown (of type **Double**)	Maximum System Drawdown
MaximumSystemDrawdownInpercentage (of type **Double**)	Maximum System Drawdown In percentage
RecoveryFactor (of type **Double**)	Recovery Factor
CarMaxDD (of type **Double**)	Car MaxDD
RarMaxDD (of type **Double**)	Rar MaxDD
NumberOfTrades (of type **Double**)	Number Of Trades
PercentOfWinners (of type **Double**)	Percent Of Winners
AnnualTurnover (of type **Double**)	Annual Turnover
TotalTradingCosts (of type **Double**)	Total Trading Costs
VolumeActivity (of type **Double**)	Volume Activity
PercentTradingCostOfVolume (of type **Double**)	Percent Trading Cost Of Volume
PercentPositiveDays (of type **Double**)	Percent Positive Days
PercentPositiveWeeks (of type **Double**)	Percent Positive Weeks
PercentPositiveMonths (of type **Double**)	Percent Positive Months
PercentPositiveYears (of type **Double**)	Percent Positive Years

AverageDailyReturn (of type **Double**)	Average Daily Return
AverageWeeklyReturn (of type **Double**)	Average Weekly Return
AverageMonthlyReturn (of type **Double**)	Average Monthly Return
UlcerIndex (of type **Double**)	Ulcer Index
UlcerPerformanceIndex (of type **Double**)	Ulcer Performance Index
KRatio (of type **Double**)	K Ratio
StandardDeviation (of type **Double**)	Standard Deviation
DownsideStandardDeviation (of type **Double**)	Downside Standard Deviation
SharpeRatio (of type **Double**)	Sharpe Ratio
SortinoRatio (of type **Double**)	Sortino Ratio
ProfitFactor (of type **Double**)	Profit Factor
PayoffRatio (of type **Double**)	Payoff Ratio
Beta (of type **Double**)	Beta
Alpha (of type **Double**)	Alpha
RSquared (of type **Double**)	R Squared
Correlation (of type **Double**)	Correlation
Fitness (of type **Double**)	Fitness value

[TOP]

1.4.9.1 Simulation Metrics

QuantShare API Documentation

Simulation Metrics Members

Fields

InitialCapital (of type **Double**)	Initial Capital
EndCapital (of type **Double**)	End Capital
NetProfit (of type **Double**)	Net Profit
NetProfitInPercentage (of type **Double**)	Net Profit In Percentage
Exposure (of type **Double**)	Exposure
NetRiskAdjustedReturn (of type **Double**)	Net Risk Adjusted Return
AnnualReturn (of type **Double**)	Annual Return
RiskAdjustedReturn (of type **Double**)	Risk Adjusted Return
AverageProfitLoss (of type **Double**)	Average ProfitLoss
AverageProfitLossInPercentage (of type **Double**)	Average ProfitLoss In Percentage
AverageBarsHeld (of type **Double**)	Average Bars Held
TotalProfit (of type **Double**)	Total Profit
NumberOfWinners (of type **Double**)	Number Of Winners
AverageProfit (of type **Double**)	Average Profit
AverageProfitInPercentage (of type **Double**)	Average Profit In Percentage
AverageBarsHeldForWinners (of type **Double**)	Average Bars Held For Winners
MaxConsecutiveWinner (of type **Double**)	Max Consecutive Winner
LargestWinner (of type **Double**)	Largest Winner
NumberOfBarsInLargestWinner (of type **Double**)	Number Of Bars In Largest Winner
TotalLoss (of type **Double**)	Total Loss

NumberOfLosers (of type **Double**)	Number Of Losers
AverageLoss (of type **Double**)	Average Loss
AverageLossInPercentage (of type **Double**)	Average Loss In Percentage
AverageBarsHeldForLosers (of type **Double**)	Average Bars Held For Losers
MaxConsecutiveLoser (of type **Double**)	Max Consecutive Loser
LargestLoser (of type **Double**)	Largest Loser
NumberOfBarsInLargestLoser (of type **Double**)	Number Of Bars In Largest Loser
MaximumTradeDrawdown (of type **Double**)	Maximum Trade Drawdown
MaximumTradeDrawdownInpercentage (of type **Double**)	Maximum Trade Drawdown In percentage
MaximumSystemDrawdown (of type **Double**)	Maximum System Drawdown
MaximumSystemDrawdownInpercentage (of type **Double**)	Maximum System Drawdown In percentage
RecoveryFactor (of type **Double**)	Recovery Factor
CarMaxDD (of type **Double**)	Car MaxDD
RarMaxDD (of type **Double**)	Rar MaxDD
NumberOfTrades (of type **Double**)	Number Of Trades
PercentOfWinners (of type **Double**)	Percent Of Winners
AnnualTurnover (of type **Double**)	Annual Turnover
TotalTradingCosts (of type **Double**)	Total Trading Costs
VolumeActivity (of type **Double**)	Volume Activity
PercentTradingCostOfVolume (of type **Double**)	Percent Trading Cost Of Volume
PercentPositiveDays (of type **Double**)	Percent Positive Days
PercentPositiveWeeks (of type **Double**)	Percent Positive Weeks
PercentPositiveMonths (of type **Double**)	Percent Positive Months
PercentPositiveYears (of type **Double**)	Percent Positive Years

AverageDailyReturn (of type **Double**)	Average Daily Return
AverageWeeklyReturn (of type **Double**)	Average Weekly Return
AverageMonthlyReturn (of type **Double**)	Average Monthly Return
UlcerIndex (of type **Double**)	Ulcer Index
UlcerPerformanceIndex (of type **Double**)	Ulcer Performance Index
KRatio (of type **Double**)	K Ratio
StandardDeviation (of type **Double**)	Standard Deviation
DownsideStandardDeviation (of type **Double**)	Downside Standard Deviation
SharpeRatio (of type **Double**)	Sharpe Ratio
SortinoRatio (of type **Double**)	Sortino Ratio
ProfitFactor (of type **Double**)	Profit Factor
PayoffRatio (of type **Double**)	Payoff Ratio
Beta (of type **Double**)	Beta
Alpha (of type **Double**)	Alpha
RSquared (of type **Double**)	R Squared
Correlation (of type **Double**)	Correlation
Fitness (of type **Double**)	Fitness value

[TOP]