1ExAlgo

API

Table of contents

API: Robots (Script)

- Data from tables
- Closing a long position
- Closing a short position
- Basic properties (candlesticks, instrument, time, etc.)
- Opening a long position
- Opening a short position
- Debugging panel, table, logs and additional columns
- Position
- Painting
- Withdrawal of applications
- ParamOptimization

Data from tables

Worksheets data

Property

TableDataModel *Tables - Data from the worksheet of limit orders and trading positions*

TableDataModulModel *AllTables - Returning the worksheet data*

Closing a long position

Description of functions for closing a long position: no return value.

Closing a long position at the previous candlestick closing price or a few seconds before it ends, depending on the current settings.
SellAtClose (int bar, IPosition pos, string comment, AddInfoToOrder add = null)

Closing a long position at a price higher or equal than a given price.
SellAtProfit (int bar, IPosition pos, double price, string comment, AddInfoToOrder add = null)

Closing a long position at the opening price of a new candlestick.
SellAtMarket (int bar, IPosition pos, string comment, AddInfoToOrder add = null)

Close a long position at a price lower or equal then a specified
SellAtStop (int bar, IPosition pos, double price, string comment, AddInfoToOrder add = null)

Close a long position via trailing return
SellAtTrallingProfit (int bar, IPosition pos, double priceActivation, double otstupFromMinMax, bool isPercentOtstupFromMinMax, string comment, AddInfoToOrder add = null)

Type	Variable	Description
IPosition	pos	the tool, due to the lack of the necessary counter
int	bar	position closing bar, always set (bar+1), except for SellAtClose - (bar)
double	price	position closing price
double	priceActivation	trailing return activation price
double	otstupFromMinMax	shift from the minimum/maximum value for trailing return
bool	isPercentOtstupFromMinMax	shift for trailing return can be calculated as a percentage or points
double	lot	position opening volume (in the case a trading robot specifies the position size calculated from the worksheet)
string	comment	trading signal name

AddInfoToOrder	add	additional parameters for the position control, do not used by default

Closing a short position

Description of functions for closing a short position. There is no return value.

Closing a short position at the previous candlestick closing price or a few seconds before it ends, by current settings.
CoverAtClose (int bar, IPosition pos, string comment, AddInfoToOrder add = null)

Closing a short position at a price lower or equal than a given price.
CoverAtProfit (int bar, IPosition pos, double price, string comment, AddInfoToOrder add = null)

Closing a short position at the opening price of a new candlestick.
CoverAtMarket (int bar, IPosition pos, string comment, AddInfoToOrder add = null)

Closing a short position at a price higher or equal than a given prive.
CoverAtStop (int bar, IPosition pos, double price, string comment, AddInfoToOrder add = null)

Closing a short position via trailing return.
CoverAtTrallingProfit (int bar, IPosition pos, double priceActivation, double otstupFromMinMax, bool isPercentOtstupFromMinMax, string comment, AddInfoToOrder add = null)

Type	Variable	Description
IPosition	**pos**	Position to be closed
int	**bar**	position closing bar, always set (bar+1), except for CoverAtClose - (bar)
double	**price**	position closing price
double	**priceActivation**	trailing return activation price
double	**otstupFromMinMax**	offset from the minimum/maximum value for trailing return
bool	**isPercentOtstupFromMinMax**	offset for trailing return can be calculated as a percentage or points

double	**lot**	position opening volume (in the case a trading robot specifies the position size calculated from the worksheet)
string	**comment**	trading signal name
AddInfoToOrder	add	дополнительные параметры для управления позицией, по умолчанию не используются

Basic properties (candlesticks, instrument, time, etc.)

List of properties available for calling a trading robot script

Type	Variable	Description
int	CandleCount	the number of candlesticks (bars) according to the traded timeframe of the main instrument.
ICandlesSeries	Candles	candlesticks by main tools
List<IStatisticDeal>	Deals	list of deals for all current robots' tools
IFinInfoInstrument	FinInfo	information about the main tools
int	IndexBar	int
bool	IsManyOpenOnOneCandle	ability to perform several operations with a candlestick using one closed trading signal: default false (disabled)
bool	IsOneDynamicTable	debug sheet is created only at the end of the test
IService	Service	additional auxiliary variables

Opening a long position

Description of functions for opening a long position. The returned value of IPosition is a created position or an existing position.

Open a long position at the closing price of the candlestick or a few seconds before it ends: by settings.
BuyAtClose(int bar, double lot, string comment, AddInfoToOrder add = null)
BuyAtClose(IFinInfoInstrument symbol, int bar, double lot, string comment, AddInfoToOrder add = null)

Open a long position at a price lower or equal than a given price.
BuyAtLimit(int bar, double price, double lot, string comment, AddInfoToOrder add = null)
BuyAtLimit(IFinInfoInstrument symbol, int bar, double price, double lot, string comment, AddInfoToOrder add = null)

Open a long position by the opening price of a new candlestick.
BuyAtMarket(int bar, double lot, string comment, AddInfoToOrder add = null)
BuyAtMarket(IFinInfoInstrument symbol, int bar, double lot, string comment, AddInfoToOrder add = null)

Open a long position by a price that is higher or equal than a given price.
BuyGreater(int bar, double price, double lot, string comment, AddInfoToOrder add = null)
BuyGreater(IFinInfoInstrument symbol, int bar, double price, double lot, string comment, AddInfoToOrder add = null)

Type	Variable	Description
IFinInfoInstrument	**symbol**	instrument to be operated on
int	**bar**	position opening bar, always set (bar+1), except for BuyAtClose - (bar)
double	**price**	position opening price
double	**lot**	position opening volume (used if the position size is specified in the robot - calculation from the script)
string	**comment**	Trading signal name
AddInfoToOrder	add	additional parameters for position management, not used by default

Opening a short position

Description of functions for opening a long position. The return value of IPosition is a created position or an existing one.

Open a short position at the closing price of the candle or a few seconds before it ends, depending on the settings.
ShortAtClose(int bar, double lot, string comment, AddInfoToOrder add = null)
ShortAtClose(IFinInfoInstrument symbol, int bar, double lot, string comment, AddInfoToOrder add = null)

Open a short position at a price equal to or higher than the specified
ShortAtLimit(int bar, double price, double lot, string comment, AddInfoToOrder add = null)
ShortAtLimit(IFinInfoInstrument symbol, int bar, double price, double lot, string comment, AddInfoToOrder add = null)

Open a short position at the opening price of a new candle.
ShortAtMarket(int bar, double lot, string comment, AddInfoToOrder add = null)
ShortAtMarket(IFinInfoInstrument symbol, int bar, double lot, string comment, AddInfoToOrder add = null)

Open a short position at a price equal to or below the specified
ShortLess(int bar, double price, double lot, string comment, AddInfoToOrder add = null)
ShortLess(IFinInfoInstrument symbol, int bar, double price, double lot, string comment, AddInfoToOrder add = null)

Type	Variable	Description
IFinInfoInstrument	**symbol**	instrument to be operated on
int	**bar**	position opening bar, always set (bar+1), except for ShortAtClose - (bar)
double	**price**	position opening price
double	**lot**	position opening volume (used if the position size is specified in the robot - calculation from the script)
string	**comment**	Trading signal name
AddInfoToOrder	add	additional parameters for position management, not used by default

Debug panel, table, logs and additional columns

Elements to help debugging the algorithm

Methods

AddLogRobot - no return value
AddLogRobot(string mess) - adds a message to the robot log; message;

ParamDebug - no return value
ParamDebug(string name, object values) - adds and updates information on the "name" key in the debug panel;
name - parameter name (key);
values - value (displayed in text format);

ClearParamDebug - no return value
ClearParamDebug() - cleaning of the information panel;

AddColumnToMainTable - return value: List<ScanerParamModel>.
virtual List<ScanerParamModel> AddColumnToMainTable() - Method that is overridden in the robot algorithm and returns the following data: the need to create additional columns in the robot panel

AddRowDynamicTable - the return value: the RowTableScaner.
AddRowDynamicTable() - The method creates a row that can be added to DynamicTableRows.

SetColumnDynamicTable - the return value: List<RowTableScaner>.
SetColumnDynamicTable() - Method that is overridden in the robot algorithm and returns the following data: the need to create columns for the debug table

Properties

List<ScanerParamModel> MainTableColumn - list of additional columns added to the robot panel.

List<RowTableScaner> DynamicTableRows - list of strings added to the debug table.

Position

Getting information about current positions. The information in the LongPos and ShortPos properties is updated immediately after the transaction is sent, MarketPosition is updated after opening or closing a position.

Properties

Type	Variable	Description
List<IPosition>	LongPos	list of current long positions
List<IPosition>	ShortPos	list of current short positions
double	MarketPosition	Current position (main instrument) in the robot

Methods

GetLongPosition - return value: IList<IPosition>
GetLongPosition() - returns a list of long positions (main instrument);
GetPositon(IFinInfoInstrument symbol) - returns a list of long positions (specified instrument);

GetShortPositions- return value: IList<IPosition>
GetShortPositions() - returns a list of short positions (main instrument);
GetShortPositions(IFinInfoInstrument symbol) - returns a list of short positions (specified instrument);

GetPositon - return value: IList<IPosition>
GetPositon() - returns a list of long and short positions (main instrument);
GetPositon(IFinInfoInstrument symbol) - returns a list of long and short positions (specified instrument);

GetLastPositon - return value: IList<IPosition>
GetLastPositon() - returns the last open position (main instrument);
GetLastPositon(IFinInfoInstrument symbol) - returns the last open position (specified instrument);

ClearPostition- return value: IList<IPosition>
ClearPostition() - Delete the position on the robot. Positions' data does not change until the next recalculation according to the algorithm; (can only be used in real trading);

ConvertPosToReal- no return value
ConvertPosToReal(IPosition pos) - Converts the current position to the real one;

ConvertPosToHistory- no return value
ConvertPosToHistory(IPosition pos) - Converts the current position to the historical one;

Painting

Methods that add additional visual elements to the chart

PlotLine - no return value
PlotLine(string nameCol, double value, UserChartPropModel chartProp) - create a horizontal line;
nameCol - line name;
value - the value: level on which the line will be plotted on a chart.
chartProp - additional parameters (color, panel number, etc.);

PlotArea - no return value
PlotArea(string nameCol, int bar, UserChartPropModel chartProp) - the background is painted over on the specified bars (all charts);
nameCol - title;
bar - bar number;
chartProp - additional parameters (color, panel number, etc.);

Plotseries - no return value
Plotseries(string nameCol, List<double> list, UserChartPropModel chartProp) - Plot data series on a chart;
nameCol - title;
list - data series;
chartProp - additional parameters (color, panel number, etc.);

PlotRangeBar - no return value
PlotRangeBar(string nameCol, List<RangeBarCol> list, UserChartPropModel chartProp) - Plot data series on a chart;
nameCol - title;
list - data series;
chartProp - additional parameters (color, panel number, etc.);

PlotCanIdes - no return value
PlotCanIdes(string nameCol, IFinInfoInstrument finInfoInstrument, UserChartPropModel chartProp) - Plot candles of the transferred instrument;
nameCol - title;
finInfoInstrument - instrument;
chartProp - additional parameters (color, panel number, etc.);

PlotPriceCanIdes - no return value
PlotPriceCanIdes(string nameCol,List<double> open, List<double> high, List<double> low, List<double> close, UserChartPropModel chartProp) - Plot candles according to specified series;
nameCol - title;
open - a series of opening prices;
high - a series of price highs;
low - a series of price lows;

close - a series of closing prices;
chartProp - additional parameters (color, panel number, etc.).

Withdrawal of orders

Description of methods (removing active orders)

Methods

Cancellation of an order "open a position"
KillOrderOpen(int bar, IPosition pos, string comment = "")

Cancellation of a stop order "open a position"
KillStopOrderOpen(int bar, IPosition pos, string comment = "")

Cancellation of an order "close a position"
KillOrderClose(int bar, IPosition pos, string comment = "")

Cancellation of a stop order "open a position"
KillStopOrderClose(int bar, IPosition pos, string comment = "")

Type	Variable	Description
int	bar	the bar where you need to cancel the order (bar+1 in all strategies where the timeframe is used)
IPosition	pos	position
string	comment	comment on canceled order

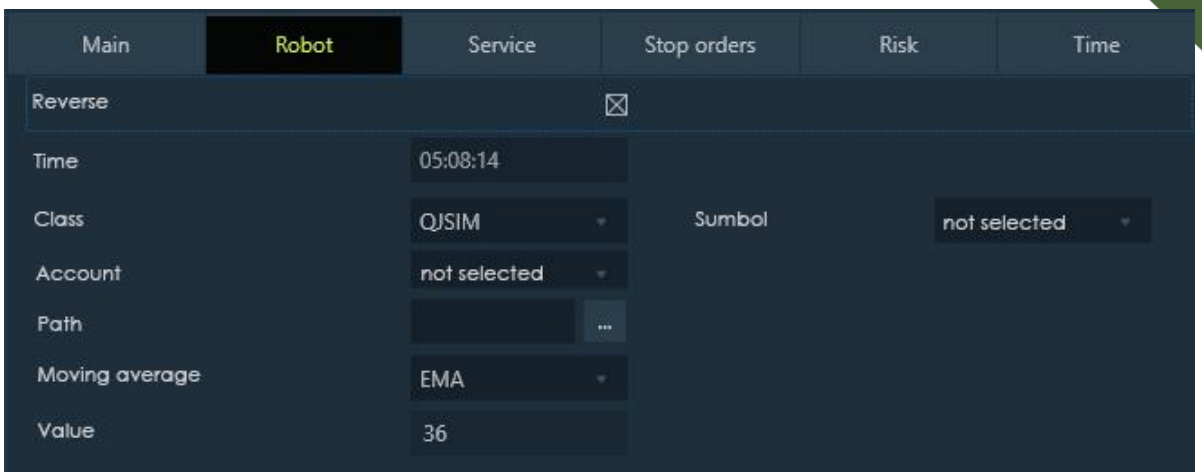
ParamOptimization

Class: create user interface

Type	Variable	Description
double	Value	fractional value
bool	ValueBool	boolean (true, false)
DateTime	ValueDateTime	time
int	ValueInt	integer value
string	ValueString	text value
List<bool>	ListBool	list of booleans
List<double>	ListDouble	list of fractional values
List<int>	ListInt	list of integers
List<string>	ListString	list of text values
string	TypeTimeFrame	timeframe type
int	TimeFrame	timeframe

Description of the ParamOptimization class, which allows the user to enter data (numeric values, boolean value, etc.)

For example:



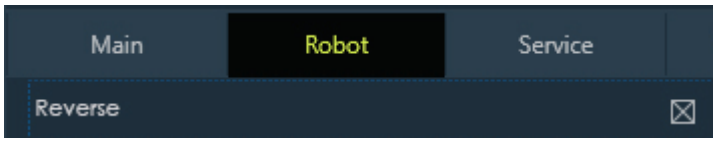
An object of this class has 6 overloads:

1. Constructor parameters for bool values
Parameters:
public ParamOptimization(bool valBool, string nameParam, string description = "")

Creating a constructor
public ParamOptimization Reverse = new ParamOptimization(true, "Reverse", "Reverse positions");

Accessing the value of a variable in code:
bool valueReverse = Reverse.ValueBool;

Result in 1ExAlgo



Type	Variable	Description
bool	valBool	boolean value
string	nameParam	parameter name
string	description	optional parameter description

2. Constructor parameters for entering time
Parameters:

public ParamOptimization(TimeSpan valueTime, string nameParam, string description = "")

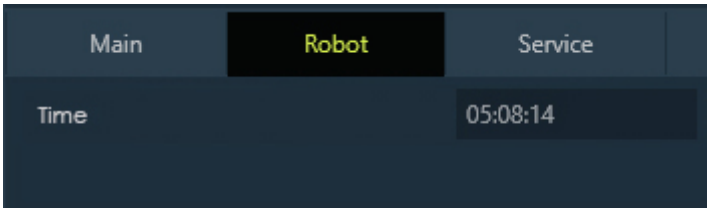
Creating a constructor

public ParamOptimization Time = new ParamOptimization(new TimeSpan(5,8,14), "Time", "Time used for calculations");

Accessing a variable in code

DateTime time = Time.ValueDateTime;

Result in 1ExAlgo



Type	Variable	Description
TimeSpan	valueTime	time
string	nameParam	parameter name
string	description	optional parameter description

3. Parameters of the constructor for using the second tool

Parameters:

public ParamOptimization(Script.EnumTypeGetIEnumerable typeGetIEnumerable, string nameParam, string description = "", bool visibleAccount = true)

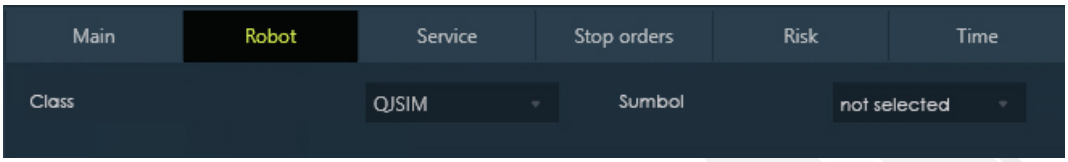
Creating a constructor

public ParamOptimization Instr = new ParamOptimization(EnumTypeGetIEnumerable.GetSeccodeList, "Using an additional tool", "Tool");

Referring to an instrument in code, for example, to the closing price of a candle

double result = Instr.FinInfo.Candles.CloseSeries[bar];

Result in 1ExAlgo



Type	Variable	Description
EnumTypeGetIEnumerable	typeGetIEnumerable	using a second instrument
string	nameParam	parameter naming
string	description	optional parameter description
bool	visibleAccount	visibility in robot settings

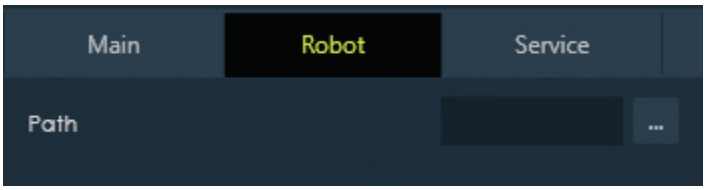
4. Constructor parameters for file upload
Parameters:

public ParamOptimization(string valueString, string nameParam, bool controlPath, string description="");

Creating a constructor
public ParamOptimization Value = new ParamOptimization("", "Path", true, "Select file path");

Accessing a variable in code
string val = Value.ValueString;

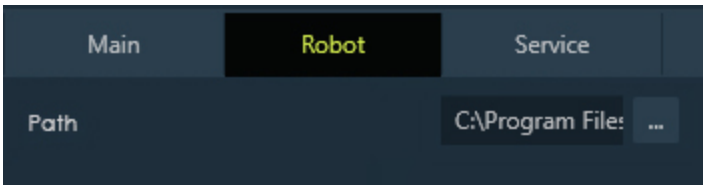
Result in 1ExAlgo



Type	Variable	Description
string	valString	variable value
string	nameParam	parameter name
bool	controlPath	allows to upload a file
string	description	optional parameter description

If the controlPath variable is set to false, then instead of selecting a file, there will be

a text field



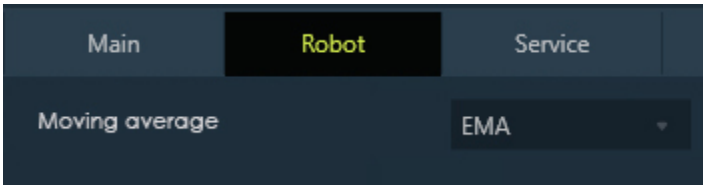
5. Constructor parameters for selecting string values
Parameters:

public ParamOptimization(string valueString, List<string> list, string nameParam, string description)

Creating a constructor
public ParamOptimization Value = new ParamOptimization("EMA", new List<string>() { "EMA", "SMA" }, "Moving Average", "Type of moving average for calculations");

Accessing the Selected Value in Code
string val = Value.ValueString;

Result in 1ExAlgo



Type	Variable	Description
string	valueString	variable value
List	list	list of string values to select
string	nameParam	parameter name
string	description	optional parameter description

6. Constructor parameters for entering double values
Parameters:

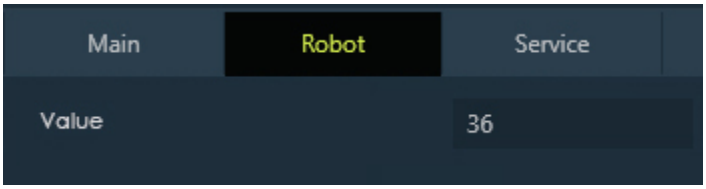
public ParamOptimization(double value, double startStep, double endStep, double stepOptimization, string nameParam, string description = "")

Creating a constructor

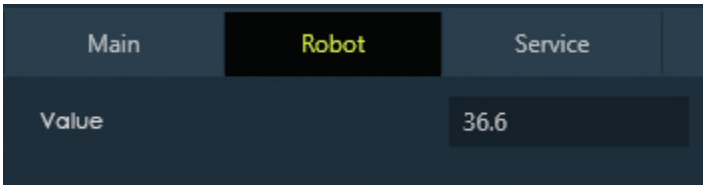
public ParamOptimization Value = new ParamOptimization(36, 0, 5, 1, "Value", "Input value for calculations");

Accessing a variable in code
double val = Value.Value;
double val = Value.ValueInt;

Result in 1ExAlgo



If the user enters the number 36.6 and there is a reference in the code: double val = Value.ValueInt, then the value is rounded up to 37, which can lead to incorrect calculations.



Type	Variable	Description
double	value	variable value
double	startStep	Start optimisation value
double	endStep	End optimisation value
double	stepOptimization	optimization step
string	nameParam	parameter name
string	description	Additional parameter description

Using of the indicators

Parameters:

public CreateIndicator(Script.EnumIndicators name, int pane, string addName)

Creating the indicator "Fractal"

public CreateIndicator FractalsDotted = new CreateIndicator(EnumIndicators.FractalsDotted, 0, "")

The class constructor has 3 parameters:

1. EnumIndicators – enumerations (enum) of indicators;
2. The number of the panel on which the indicator is displayed;
3. Name.

Interaction with the indicator in code

```
var Up = FractalsDotted.param.LinesIndicators[0].LineParam[0].Value
var Down = FractalsDotted.param.LinesIndicators[1].LineParam[0].Value
```

Appeal to the upper fractal

```
double upValue = FractalsDotted.param.LinesIndicators[1].PriceSeries[bar]
```

Appeal to the lower fractal

```
double downValue = FractalsDotted.param.LinesIndicators[0].PriceSeries[bar]
```

The ChangeParamOptimization method allows you to change the value specified in ParamOptimization in the code

For example, let's declare ParamOptimization with a value of "double" and change it using the function above.

```
public ParamOptimization Value = new ParamOptimization(36, 5, 5, 1, "Value");

Lets assign a new value and change it.
Value.ValueInt = 5;

ChangeParamOptimization(Value);
```

Thus, the value 36 will be replaced by the value 5.

API: Classes and Interfaces

- AddInfoToOrder
- CandleModel
- ClasterCandleModel
- ClasterDetailModel
- EnumIndicators
- EnumScanerParamType
- EnumStatusOrders
- EnumTypeLine
- ICandlesSeries
- IComissionSettingModel
- IDeals
- IFinInfoInstrument
- IGlassQuotation
- IMoneyFutures
- IMoneyShares
- IOrders
- IPosition
- IPositionFutures
- IPositionShares
- ISecurity
- IService
- IStatisticDay
- IStatisticDeal
- IStop
- RangeBarCol
- RowTableScaner
- ScanerParamModel
- TableDataModel
- TableDataModulModel
- UserChartPropModel

AddInfoToOrder

Additional parameters for position control

Variable

Type	Variable	Description
bool	IgnoreGapForOpenLimit	for testing, the execution takes place at the specified price, the opening gap is not taken into account
bool	IsAlwaysClose	the deal must be closed regardless of the price
bool	IsAlwaysOpen	the deal must be opened regardless of the price
bool	IsHistorical	the deal should open as historical
string	Info	additional user information
double	IndividualSlippage	individual slipping

CandleModel

Candle (bar) or parameters of an impersonal (depersonalised?) transaction

Type	Variable	Description
double	Close	closing price
double	High	maximum price
bool	IsOperBuy	"Buy" operation, otherwise Sell
double	Low	minimum price
int	Oi	open interest
double	Open	opening price
DateTime	TradeDateTime	candle time (trade)
double	VolBuy	Buy volume
double	VolSell	Sell volume
double	Volume	overall volume

ClasterCandleModel

Clusters by time

Type	Variable	Description
double	BuyVol	Buy volume
double	SellVol	Sell volume
double	Delta	delta = Buy volume - Sell volume
List<ClasterDetailModel>	ClasterDetails	Detailed information on the cluster
double	BuyCount	amount of "Buy" trades
double	SellCount	amount of "Sell" trades

ClasterDetailModel

Detailed information about clusters by time

Type	Variable	Description
double	BuyVol	Buy volume
double	Price	price
double	SellVol	Sell volume

EnumIndicators

List of indicators that are programmed in 1ExAlgo

Name	Full name	lines
Ac	Acceleration/Deceleration indicator	1 - AC
Ad	Williams' Accumulation/Distribution Indicator	1 - AD
Adx	Average Directional Movement Indicator	1 - ADX 2 - DI+ 3 - DI-
Adx2	Indicator of average directional movement - as in QUIK terminal	1 - ADX 2 - DI+ 3 - DI-
Ama	Adaptive Moving Average	1 - AMA
Alligator	"Alligator" indicator	1 - jaws 2- teeth 3 - lips
Ao	Awesome Oscillator	1 - AO
Atr	Average True Range	1 - ATR
BollinderBands	Bollinger bands	1 - middle line 2 - top line 3 - bottom line
BearsPower	Power of Bears (sellers)	
BullsPower	Power of Bulls (buyers)	
ChaikinVolatility	Chaikin Volatility	
Cci	Commodity Channel Index	1 - CCI

ChaikinOscillator	Chaikin Oscillator	1 - Chaikin oscillator
CumulVolume	accumulated volume	1 - accumulated volume
DiPlus	Di+	1 - DI+
Envelopes	Envelopes	
Fractals	Bill Williams fractals (solid lines)	1 - upper fractal (line) 2 - lower fractal (line) 3 - upper fractal (dots) 4 - lower fractal (dots)
FractalsDotted	Bill Williams fractals (dots)	1 - upper fractal (dots) 2 - lower fractal (dots)
Ichimoku	Ichimoku indicator	1 - Tenkan 2 - Kijun 3 - Senkou Span1 4 - Senkou Span2
Macd	Moving Average Convergence/Divergence indicator	1 - Macd 2 - Signal line
MFI	Money Flow Index	
Momentum	Market speed indicator	1 - Momentum
MovingAvarage	moving average	1 - moving average
Osma	Moving Average of Oscillator	1 - MACD 2 - signal line 3 - Osma histogram
ParabolicSar	Parabolic Stop and Reverse (SAR)	1 - SAR 2 - Previous SAR value
Pivot	Pivot	1 - upper pivot (line) 2 - lower pivot (line) 3 - upper pivot (point) 4 - lower pivot (point)

PriceChannel	price channel	1 - upper channel 2 - lower channel
Rsi	Relative Strength Index	1 - RSI
Stochastic	Stochastic Oscillator	1 - %K 2 - %D
Trix	Triple Exponential Moving Average	1 - TRIX
VolBuySell	Indicator	1 - Buy volume 2 - Sell volume
Volume	Volume	1 - volume
Vwap	Volume Weighted Average Price	1 - VWAP
WilliamsPercentsR	Williams' %R, Williams Percent Range	1 - % Range

EnumScanerParamType

Types of possible column values in the debug table

Value	Description
typeBool	Bool data type
typeDateTime	DateTime data type
typeDouble	Double data type
typeInt	Int data type
typeString	String data type
typeTime	Time data type

EnumStatusOrders

Status of orders to open and close positions

Value	Description
Null	The Exchange returned an error on the submitted order
Active	Order is active
Performed	Order fulfilled
Cancel	Order canceled
AddToTs	The order was sent to the Exchange, after the activation of the Stop-order
RemoveFormTs	The order was deleted after the activation of the Stop-order
Unknow	The order has been sent, but the answer has not yet come from the exchange

EnumTypeLine

Line types for plotting

Value	Description
Line	Line
Dotted	
Histogram	bar graph
StepLine	Step line
InvertalTriangle	inverted triangle
LineDash	dot-dash line
LineDot	dotted line
StepLine	walking line, changes at 90 degrees
Triangle	triangle

ICandlesSeries

Candle data series

Type	Variable	Description
int	CandleCount	number of candles (bars) in a series
List<ClasterCandleModel>	ClasterTimeSeries	time cluster
List<double>	CloseSeries	closing price
List<DateTime>	DateTimeCandle	candle (bar) time
List<double>	HighSeries	maximum price
List<double>	LowSeries	minimum price
List<double>	MedianSeries	average price (high+low)/2
List<int>	Oi	open interest
List<double>	OpenSeries	opening price
List<double>	TypicalSeries	average price (high+low+close)/3
List<double>	VolBuy	Buy volume
List<double>	VolSell	Sell volume
List<double>	Volume	overall volume

IComissionSettingModel

Instrument commission

Type	Variable	Description
int	Accuracy	commission calculation accuracy
double	Comission	commission
double	MaxComis	maximum commission
string	MaxTypeComis	type of maximum commission (do not use, percentage of the transaction)
double	MinComis	minimum commission
string	TypeComis	commission type (percentage, fixed)

IDeals

Deals table

Type	Variable	Description
string	Account	account
string	ClassCode	class code
string	ClientCode	Client code
string	Comment	comment
DateTime	DateTrade	transaction date
string	NumberTrade	transaction number
string	Operation	operation "Buy" / "Sell"
string	Order	order number
double	Price	price
double	Quantity	amount
string	Symbol	instrument code
double	Volume	volume

IFinInfoInstrument

Instrument: Financial information

Type	Variable	Description
string	Account	account
List<CandleModel>	AllTrades	list of all trades of the day
ICandlesSeries	Candles	a series of candlestick data of the current timeframe. If the strategy uses two or more timeframes, then the larger timeframe will be split and the number of candles will be brought to the base timeframe.
string	ClassCode	class code
string	ClientCode	Client code
IComissionSettingModel	Comission	instrument commission
DateTime	LastTimeUpdateGlassUtc	last server time for order book update
int	PeriodTimeFrameDecompress	compression timeframe period
int	PeriodTimeFrameSource	source timeframe period
int	Slippage	slippage

List<IGlassQuotation>	QuotationsBuy	Depth of Market (DOM) quotations for Buying
List<IGlassQuotation>	QuotationsSell	Depth of Market (DOM) quotations for Selling
string	Seccode	string
ISecurity	Security	instruments' financial information from the table of current parameters
DateTime	SendTimeUpdateGlassLocal	local time of sending a request for quotations by DOM
string	TypeTimeFrameDecompress	compression timeframe type
string	TypeTimeFrameSource	source timeframe type

Methods

public int GetNumberCandleBigFrame(int bar, int shift)

Get the bar number of the higher timeframe when two or more different timeframes are used in the strategy.

the current bar of the lower base timeframe on which the strategy is traded

shift relative to the specified candle on a larger timeframe. With a shift of 0, the last formed candle is displayed relative to the specified candle, or the specified candle if it has already been formed.

In case of accessing to a non-existent candle, the value "-1" will be returned.

IGlassQuotation

Depth of Markets (DOMs) quotations

Type	Variable	Description
double	BuyQty	volume to buy
double	Price	price
double	SellQty	volume to sell

IMoneyFutures

Futures cash table

Type	Variable	Description
string	Account	account
double	ExchangeFee	the amount charged by the exchange committee for conducting exchange transactions. FORTS market parameter.
double	CurrentEmptyOpen	amount of collateral reserved for open positions, in monetary terms
double	CurrentEmptyOrder	amount of collateral reserved for active bids, in monetary terms
double	CurrentEmptyPosition	the total cash collateral reserved for open positions and trading operations of the current session. For the RTS Standard market, only positions in major spot assets are taken into account*
double	Dohod	double
double	LastLimitOpenPosition	limit of open positions for all instruments of the previous trading session in cash
double	LimitOpenPosition	the current limit of open positions for all instruments in cash For the RTS Standard market - displays the limit of the purchase of spot assets
double	PlanEmptyPosition	planned net positions for all instruments in monetary terms Corresponds to the Free Margin parameter of the FORTS market.
string	TypeLimit	limit type for the FORTS market: "Cash" - the value of cash provision
double	VariableMarga	variation margin for client positions, for all instruments

IMoneyShares

Stock (equity) cash table

Type	Variable	Description
string	Account	account
double	Balance	balance
string	ClientCode	Client code
string	Group	group
string	LimitKind	limit type
double	OpenBalance	opening balance
string	Currency	currency

IOrders

Order table

Type	Variable	Description
string	Account	Account
bool	AnotherServer	the order is placed on another server, assigned when trying to cancel the order
double	Balance	Order remnant
string	ClassCode	class code
string	ClientCode	Client code
string	Comment	comment
DateTime	DateTimeKill	Cansellation time
string	Id	unique identificator
string	Number	order number
string	Operation	operation "Buy" / "Sell"
double	Price	price
double	Quantity	number
string	Status	order status (Undefined, Active, Completed, Canceled)
string	Symbol	Instrumet code

DateTime	Time	order placement time

IPosition

Description of the IPosition interface.

Type	Variable	Description
string	Account	string
double	CalcProfit	calculated values of stop profit, when setting stops from the robot form
double	CalcStop	calculated values of the stop order, when setting stops from the robot form
double	CalcTrallingStopActive	calculated values stop position activation price, when setting stops from the robot form
string	ClassCode	class code
string	ClientCode	Client code
int	EntryBar	position opening bar
bool	EntryIsActive	order to open a position is active
string	EntryNameSignal	signal to open a position
EnumStatusOrders	EntryOrderStatus	position opening order - status
double	EntryPrice	signal price for opening a position
DateTime	EntryTime	position opening time
DateTime	EntryTimeLocal	local position opening time
int	ExitBar	Position closing bar

bool	ExitIsActive	an order to close a position is active
EnumStatusOrders	ExitOrderStatus	position closing order status
double	ExitPrice	position closing signal price
DateTime	ExitTime	position closing time
DateTime	ExitTimeLocal	local time of position closing
string	Info	additional user information
bool	IsDelPos	position deleted
bool	IsLong	long position
bool	IsReal	the position is real, otherwise historical
bool	IsShort	short position
double	Pos	current position
double	QuantityOpen	closing quantity
double	QuantityClose	quantity to open
IStatisticDeal	Statistic	position statistics
string	Symbol	instrument code

IPositionFutures

Futures positions table

Type	Variable	Description
double	ActiveBuy	in Buy orders
double	ActiveSell	in Sell orders
double	Balance	balance
string	ClientCode	Client code
double	CurrentLongPos	current long positions
double	CurrentShortPos	current short positions
double	EnterEmptyPos	incoming net positions
string	Symbol	instrument code
double	VariableMarga	variation margin

IPositionShares

Futures cash table

Type	Variable	Description
string	Account	account
double	Balance	balance
string	ClientCode	Client code
double	EnterOst	incoming balance (remaining)
DateTime	LastTimeUpdate	last update time
string	NameSymbol	instrument name
string	Symbol	instrument code

ISecurity

Instrument data (stock, bond, futures, currency, etc.) displayed in the table of current parameters.

Type	Variable	Description
double	Accuedint	accumulated coupon income
int	Accuracy	precision, amount of decimal places
string	BaseActive	basic asset
double	Bid	best buy price
string	ClassCode	securities' class
double	ClosePrice	closing price
int	CouponPeriod	coupon duration
double	CouponValue	coupon size
DateTime	DateExpire	expiration date
int	DaysToMatDateBond	days to maturity of the bond
double	Duration	duration
double	GoBuy	buyer's warranty
double	GoSell	seller's warranty
bool	IsCrypto	cryptoInstrument
string	Isin	digital instrument code

bool	IsTrade	instrument trading allowed
double	LastChangePercent	percentage change from the previous session's closing price
string	LastNumberTrade	last deal number (tick)
double	LastPrice	last trade price
double	LotSize	lot size
DateTime	MatDate	maturity date
double	MaxAmount	maximum volume
double	MaxPrice	the maximum possible price for the instrument, for placing an order
double	MinAmount	minimum volume
double	MinNational	minimum transaction amount
double	MinPrice	the minimum possible price for the instrument, for placing an order
double	MinStep	price step
DateTime	NextCoupon	next coupon payment date
double	Offer	best selling price
double	OpenPrice	opening price
double	PointCost	price step cost
string	QuotesBasis	price type

string	Seccode	instrument code
double	SecFaceValue	bond face value
string	SecType	instrument type
string	ShortName	short name of the instrument
string	Status	session status
double	Strike	option strike
double	TheorPrice	theoretical option price
DateTime	TimeLastChange	last modification (change) time
string	TradingStatus	trade status
string	TypeOption	option type
string	TypePriceBond	bond price type
double	Volatility	option volatility
double	VolatToday	cash turnover
double	Yield	profit rate
int	LotSizeAccuracy	lot size accuracy (cryptocurrency only)

IService

Description of additional variables

Type	Variable	Description
string	Account	account
string	ClientCode	Client code
bool	IsCloseByCloseCandleBeforeAnySeconds	in the settings of the robot - to close the position a few seconds before the end of the candle formation
bool	IsOpenByCloseCandleBeforeAnySeconds	in the settings of the robot - to open a position a few seconds before the end of the candle formation
bool	IsParamChange	parameters have been changed in the robot (it is necessary if non-essential parameters are changed in the robot and "Do not recalculate the robot algorithm" is set in the settings)
bool	IsPretradeEmulation	Pre-trade emulation in progress
bool	IsRealAction	real trading, otherwise - test
bool	IsTypeRobotAdviser	real trading in automatic mode
DateTime	ServerTime	server time
int	TimeFrame	timeframe time
int	Service.TimeFrameSource	compressed timeframe
string	TypeTimeFrame	type of timeframe used for trading in the robot. (Do not use, Tick, Second, Minute, Day, Week, Month)
string	TypeTimeFrameSource	the timeframe type from which the TimeFrame is compressed. (Do not use, Tick, Second, Minute, Day, Week, Month)
string	TypeTorg	Trade Type (Long, Short, Long and Short)

IStatisticDay

Description of the IStatisticDay interface (statistics for the day)

Type	Variable	Description
double	AvaregeLoss	average loss per day in cash
double	AvaregeLossPercent	average loss per day in percent
double	AvaregeLossPoint	average loss per day in points
double	AvaregeProfit	average profit per day in cash
double	AvaregeProfitLoss	Average cumulative profit per day in cash
double	AvaregeProfitLossPercent	Average cumulative profit/loss per day in percent
double	AvaregeProfitLossPoint	Average cumulative profit per day in points
double	AvaregeProfitPercent	average profit per day in percent
double	AvaregeProfitPoint	average daily profit in points
double	ChangeMoney	account change
double	ChangeMoneyPercent	account change (%)
int	CountDeals	number of transactions per day
double	CurProfitLossDay	current profit on all trades per day, including ones opened in cash
int	CurStreakDealLoss	current series of unprofitable trades for the day
int	CurStreakDealProfit	loss in the current series of unprofitable trades per day

double	CurStreakLoss	loss in the current series of unprofitable trades per day (%)
double	CurStreakLossPoint	loss in the current series of unprofitable trades per day in points
double	CurStreakProfit	profit in the current series of profitable trades per day
double	CurStreakProfitPoint	profit in the current series of profitable trades per day in
double	FactorRecovery	recovery factor per day
double	GrossLoss	total loss per day (%)
double	GrossLossPoint	total loss per day (cash)
double	GrossProfit	total daily profit (%)
double	GrossProfitPoint	total loss per day (cash)
int	LossDeals	unprofitable trades per day
double	LossDealsPercents	unprofitable trades per day (%)
double	MaxDrownDown	maximum drawdown per day (%)
double	MaxDrownDownPoint	maximum drawdown per day in points
double	MaxProfit	maximum daily profit (%)
double	MaxProfitPoint	maximum daily profit in points
int	MaxStreakDealLoss	the maximum number of unprofitable trades in a row per day
int	MaxStreakDealProfit	the maximum number of profitable trades in a row per day
double	MoneyEnterOnAccount	money in the account at the beginning of the day
double	MoneyOnAccount	money in the account

double	NetProfitLoss	total profit for the day without open positions (%)
double	NetProfitLossPoint	total profit for the day without open positions in points
int	ProfitDeals	Profitable trades per day
double	ProfitDealsPercents	Profitable trades per day (%)
double	ProfitFactor	Profit factor per day
double	RealAvaregeLoss	actual average loss per day in money without open positions
double	RealAvaregeLossPercent	actual average loss per day in percent without open positions
double	RealAvaregeLossPoint	actual average loss per day in points without open positions
double	RealAvaregeProfit	actual average profit per day in cash without open positions
double	RealAvaregeProfitLoss	actual average profit/loss per day without open positions in the money
double	RealAvaregeProfitLossPercent	actual average profit/loss per day without open positions in percent
double	RealAvaregeProfitLossPoint	actual average profit/loss per day without open positions in points
double	RealAvaregeProfitPercent	actual average profit per day (%) without open positions
double	RealAvaregeProfitPoint	actual average profit per day in points without open positions
double	RealCurProfitLossDay	actual current profit/profit on all trades per day, including open ones
double	RealCurStreakLoss	actual loss in the current series of unprofitable trades per day in percent
double	RealCurStreakLossPoint	actual loss in the current series of unprofitable trades per day in money
double	RealCurStreakProfit	actual profit in the current series of profitable trades per day in percent
double	RealCurStreakProfitPoint	actual profit in the current series of profitable trades per day in money
double	RealGrossLoss	actual total profit/loss for the day (%)

double	RealGrossLossPoint	actual total profit/loss for the day (cash)
double	RealGrossProfit	actual total profit for the day as a (%)
double	RealGrossProfitPoint	actual total daily profit (cash)
double	RealMaxDrownDown	actual maximum drawdown per day (%)
double	RealMaxDrownDownPoint	actual maximum drawdown per day (cash)
double	RealMaxProfit	actual maximum daily profit (%)
double	RealMaxProfitPoint	actual maximum daily profit (cash)
double	RealNetProfitLoss	actual total profit for the day (%)
double	RealNetProfitLossPoint	actual total profit for the day without open positions (cash)
double	SlippageClose	average slippage per close
double	SlippageOpen	average slipage per opening
List<IStatisticDeal>	TradeDealsCloseList	closed trades
List<IStatisticDeal>	TradeDealsOpenList	open trades

IStatisticDeal

Statistical information about the position

Type	Variable	Description
int	BarEnter	position opening bar
int	BarExit	closing bar
int	BarHoldPosition	holding position in bars = BarExit-BarEnter
string	ClassCode	Instrument class code
double	Comission	instrument commission
DateTime	DateTimeEnter	position opening time
DateTime	DateTimeExit	position closing time
double	GrossProfitLoss	accumulated profit/loss
bool	IsDealClose	closing position
bool	IsHistoricalDeal	historical position
bool	IsLong	long position
double	Mae	The maximum unfavorable deviation (maximum floating loss) during the time the position was held.
double	MaePoint	Maximum Adverse Excursion points. The maximum unfavorable deviation (maximum floating loss) during the time the position was held.
double	Mfe	The maximum favorable deviation (maximum floating profit) during the time the position was held.

double	MfePoint	Maximum Favorable Excursion points. The maximum favorable deviation (maximum floating profit) during the time the position was held.
int	NumberDeal	trade number on the exchange
double	PriceEnter	signal price for opening a position
double	PriceEnterCurrent	current position opening price
double	PriceExit	position closing signal price
double	PriceExitCurrent	current position closing price
double	ProfitLoss	Profit / Loss
double	ProfitLossPercents	Profit / Loss (%)
double	ProfitLossPoint	profit/loss in points
double	Quantity	position size
double	RealGrossProfitLoss	real accumulated profit/loss
double	RealMae	Maximum Adverse Excursion (only in live trading). The maximum unfavorable deviation (maximum floating loss) during the time the position was held.
double	RealMaePoint	Maximum Adverse Excursion pips (only for live trading). The maximum unfavorable deviation (maximum floating loss) during the time the position was held.
double	RealMfe	% Maximum Favorable Excursion (only in live trading). The maximum favorable deviation (maximum floating profit) during the time the position was held.

double	RealMfePoint	Maximum Favorable Excursion pips (only in live trading). The maximum favorable deviation (maximum floating profit) during the time the position was held.
double	RealPriceEnter	actual position opening price
double	RealPriceExit	actual position closing price
double	RealProfitLoss	profit/loss (only in real trading) in points
double	RealProfitLossPercents	profit/loss % (only in live trading)
double	RealProfitLossPoint	profit/loss points (only in live trading)
double	RealSlippageEnter	actual slippage in a trade to open
double	RealSlippageExit	actual slippage in the trade to close
double	RealSlippagePercentEnter	actual slippage in a trade to open (%)
double	RealSlippagePercentExit	actual slippage in a trade to close (%)
DateTime	SignalDateTimeEnter	position opening time
DateTime	SignalDateTimeExit	position closing time
string	SignalEnter	position opening signal
string	SignalExit	position closing signal
string	Symbol	instrument code
int	TimeHoldPosition	position holding time in seconds
string	Info	user information that is specified when opening a position and is changed during its maintenance

IStop

Stop-order table

Type	Variable	Description
string	Account	account
bool	AnotherServer	stop order placed on another server
double	Balance	current balance
string	ClassCode	class code
string	ClientCode	Client code
string	Comment	comment
DateTime	DateTimeKill	stop-order removal time
string	Id	unique transaction number
bool	IsPercentOtstupMaxMin	offset from the minimum / maximum for trailing profit (%)
bool	IsPercentSpread	spread from min/max for trailing profit (%)
string	Number	stop-order number
string	NumberOrder	order number from the stop-order
string	Operation	operation "Buy" / "Sell"
double	OtstupMaxMin	offset from the minimum / maximum for trailing profit
double	Price	price for the linked order (stop-limit and profit)
double	Quantity	the amount of lots/contracts in the stop-order

string	Result	server response when a stop order is activated (Rejected by the broker, Placed in the trading system, Executed by stop, Executed by profit, Did not pass limit control)
double	Spread	protective spread for a stop-order
string	Status	stop order status (Undefined, Active, Executed, Canceled, Order placed in TS, rejected in TS)
double	StopPrice	stop-limit price
string	Symbol	instrument code
DateTime	Time	date and time of the stop order
double	TpActivationPrice	trailing profit activation price
string	TypeStop	stop order type (Stop Limit, Trailing Profit, Stop Limit and Profit, Trailing Profit and Stop Limit)

RangeBarCol

RangeBar class (for plotting)

Type	Variable	Description
System.Drawing.Color	ColorRangeBar	Colour
int	NumberBar	the number of the candle on which the RangeBar will be plotted
double	PricePoint1	first point of RangeBar
double	PricePoint2	second point of RangeBar

RowTableScaner

Table row. Each table row contains a list of columns and their values

Property
List<ScanerParamModel> ColumnTable

ScanerParamModel

Class descriptions for storing cell data (cell of a debug table or of an additional column of the robots panel)

Type	Variable	Description
string	ColorBackground	Colour
string	Name	Name
CfgSourceEts.EnumScanerParamType	TypeValue	data type
double	Value	double value
bool	ValueBool	bool value
int	ValueInt	int value
string	ValueString	string value
bool	Visible	visibility

TableDataModel

Description of the class according to the tables related to the account on which the robot works.

Type	Variable	Description
IMoneyFutures	MoneyFutures	Information about cash limits for futures, if the robot uses a stock account, this value is null
IMoneyShares	MoneyShares	Information about cash limits for shares, if the robot uses a share account, this value is null
IPositionFutures	PositionFutures	Information about positions in futures, if the robot uses a stock account, this value is null
IPositionShares	PositionShares	Information about positions in stocks, if the robot uses a stock account, this value is null
List<ISecurity>	Securuties	Information on all instruments loaded in 1ExAlgo

TableDataModulModel

All data from all tables

Type	Variable	Description
ObservableCollection<IDeals>	Deals	trades table
ObservableCollection<IMoneyFutures>	MoneyFutures	futures limits table
ObservableCollection<IMoneyShares>	MoneyShares	stock limits table
ObservableCollection<IOrders>	Orders	orders table
ObservableCollection<IPositionFutures>	PositionFutures	futures positions table
ObservableCollection<IPositionShares>	PositionShares	stock positions table
ObservableCollection<ISecurity>	Securuties	table of current parameters
ObservableCollection<IStop>	Stops	stop-orders table

UserChartPropModel

Graph Options

Type	Variable	Description
System.Drawing.Color	ColorLine	Colour
int	DeepUpdate	indicator replotting depth when adding a new candle
bool	IsNotPlotSeries	plot on a chart
int	NumberPanel	number of the panel on which it is plotted
byte	Transparency	transparency (0 - transparent, 100 - opaque) Или non-transparent, или мб solid
CfgSourceEts.EnumTypeLine	TypeLine	line type
int	Width	line thickness