

Основное описание.

Описание блоков (Block).

Блок - это строительный камень Visual Forex, который производит одну или несколько операций. Блоки могут быть трех типов: **Component** (компоненты), **Indicator** (индикаторы) и **Strategy** (стратегии). Блоки главных частей любой стратегии. Каждый блок имеет уникальный алгоритм, например, блок может нарисовать индикатор, сравнить два показателя или совершить сделки. Блок может представлять сложную стратегию или выполнить простое действие.

Чтобы добавить блок в стратегию, выберите ее из меню справа и перетащите его в рабочее пространство.

Каждый блок имеет входные секции (все параметры задаются) и выходные секции, которые определяют результаты работы блока. Блок станет зеленым, когда входной и исходной информации достаточно для работы данного блока и он будет исправно использоваться в стратегии. Если что-то отсутствует, то блок будет оставаться красным.

Описание Индикаторов (Indicator).

Indicator - это тип категории блок. Индикатор возвращает по крайней мере одно выходное значение, которое должно быть заполнено обязательно. Каждый Индикатор имеет по крайней мере три обязательных входных параметров: **Instrument** (инструмент), **Period** (период) и **shift** (сдвиг целочисленного значения) и эти значения должны быть заполнены. В большинстве случаев они также имеют свои собственные уникальные параметры. **Instrument** - измерительный инструмент, индикатор баз данных на расчет; **Period** - период агрегации свечей/баров, которые использует индикатор; и **Shift** - числовой ряд от последней текущей свечи. Последняя текущая свеча - сдвиг имеет индекс 0, предыдущая свеча - сдвиг имеет индекс 1 и так далее.

В случае, если индикатор имеет несколько выходных значений и вам нужны эти значения в другом блоке, то просто создайте, для каждого выходного значения свою отдельную переменную, которая будет содержать данные индикатора. Затем, вы можете перетащить эту переменную туда, куда это необходимо.

Индикатор может также иметь параметры - **OfferSide** и **AppliedPrice**, эти параметры являются специфическими и в большинстве случаев не являются обязательными. В **OfferSide** определяет входные данные сторон - **bid** или **ask**. И выбирает цену или формулу цены для ввода данных индикатора.

Описание Компонентов (Component).

Component (Компонент) - это тип блока, который содержит различные базовые функции. Блоки компонентов поддерживаются и разлизываются в dukascopy.

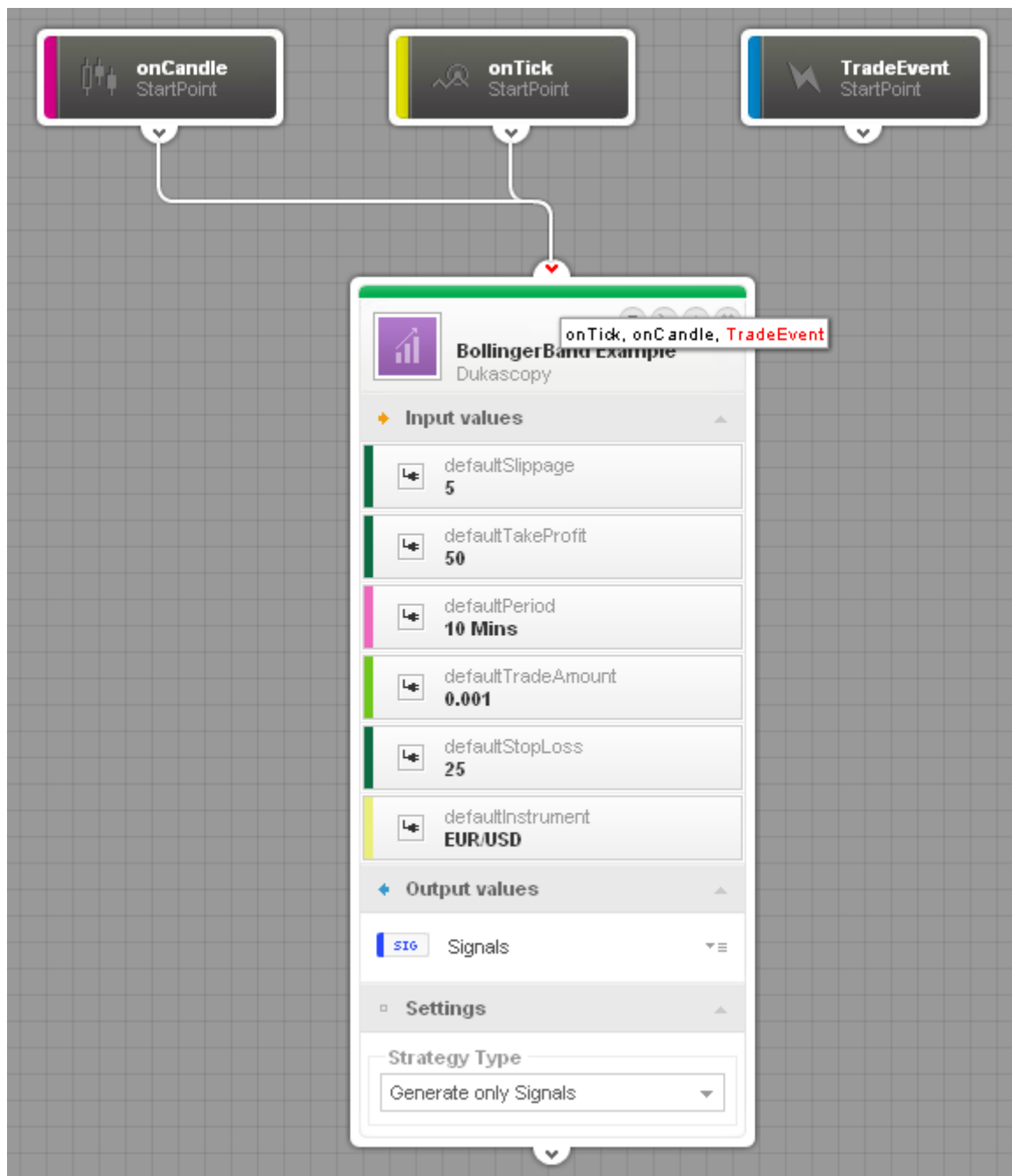
Описание Стратегии (Strategy).

Strategy (Стратегия) - это тип блока, который выводит в отдельные стратегии. Существуют три источника стратегий, которые позволяют стратегии быть доступными в виде блока в Visual jforex:

- 1) от конкурса - все публичные стратегии становятся доступными в качестве компонентов в Visual jforex от, они доступны для всех пользователей, даже для анонимных пользователей;
- 2) мои стратегии - содержит все текущие пользовательские стратегий, построенные пользователем, эти стратегии компоненты доступны только пользователю, который их сохранил.
- 3) опубликованные стратегии - стратегии, которые были разработаны и опубликованы специально для Visual jforex от как "компонент стратегии".

Стандартный блок стратегии, если он создается в Visual JForex содержит, по умолчанию Global (Global глобальные - это переменный параметр, который определяет, что переменная будет параметром тогда стратегия работает) параметры: **defaultInstrument**, **defaultPeriod**, **defaultSlippage**, **defaultTradeAmount**, **defaultStopLoss** и **defaultTakeProfit**.

Блок Стратегии может потребовать блокировать все или несколько начальных точек (**StartPoint**) flowsINs. Это можно понять наведением мышки на блок стратегии стратегию блока плавный или плавный цвет, см. пример:



Блок стратегии может использовать 2 возможности, которые можно переключать в настройках:

1) независимые - в данном случае стратегия торгует (открывает и закрывает) собственные сделки и делать то, что должна делать.

2) генерировать только сигнал - в этом случае, стратегия будет только вычислять свою собственную логику и вместо торговли (открытия или закрытия), она генерирует торговые сигналы. Поэтому, стратегия **Output value** (выходное значение) представляет коллекцию торговых сигналов, одну 1 тик стратегии блок может генерировать несколько торговых сигналов (действий). Эти сигналы можно использовать в родительской стратегии для их анализа и использования, см. пример:

BollingerBand Example
Dukascopy

Input values

- defaultSlippage: 5
- defaultTakeProfit: 50
- defaultPeriod: 10 Mins
- defaultTradeAmount: 0.001
- defaultStopLoss: 25
- defaultInstrument: EUR/USD

Output values

- Signals: bollingerSignalList

Settings

Strategy Type: Generate only Signals

Loop Viewer
Dukascopy

Input values

- Array of Data: bollingerSignalList

Output values

- Current Signal: currentSignal

If
Dukascopy

Input values

- First Last Ask Candle.Candle
- Second Last Ask Candle.Candle

Output values

If
Dukascopy

Input values

- First currentSignal.Signal Type
- Second Position Buy

Output values

Open at Market
Dukascopy

Trade Command

BUY

Input values

Output values

Settings

Описание Типа Данных (Data Type).

Тип данных - это классификация определений одного из различных типов данных, которые используются в Visual jforex как типы данных переменных. В Visual jforex можно найти следующие типы данных:

Integer - Целое число

Date and Time - Дата и время (длительность)

Double - Вещественный

Boolean - Булево

String - Строка

Messagetype - Сообщение

OfferSide – Сторона предложения

Instrument - Инструмент

Period - Период

Command - Команда

State – Состояние

AppliedPrice – приложение цены

SignalType - сигнал

MAType – скользящая средняя

Position - позиция

Message - Сообщение

Tick - Тик

Candle - Свеча

Signal - Сигнал

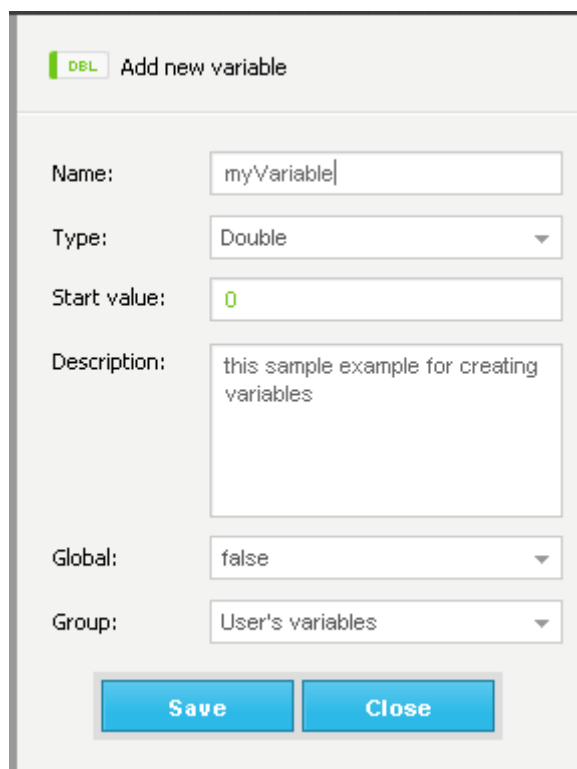
Positions - позиции

Candles - Свечи

Signals - Сигналы

Тип данных переменные (Variable). Описание типа.

Переменные – переменная, это символическое имя, данное некоторой известной или неизвестной величине, или информации, для того, чтобы имя, использовалось независимо от информации, которую он представляет. Каждая переменная должна иметь уникальное имя, значение и тип данных. Как вы можете видеть на изображении, при добавлении переменной необходимо определить по крайней мере два поля: **Name** (Название) и **Type** (тип). Эта переменная может иметь **Start value** (начальное значение) и **Description** (описание). Описание используется только для информации пользователя, оно не имеет никаких других функций. Также пользователь может определить группу, к которой будет относиться данные, для этого существует параметр **Global**, который может быть false (ложным) или true (истинным). Если параметр **Global** истина - это делает эту переменную видимой в качестве параметра стратегии, когда она запущена.



DBL Add new variable

Name: myVariable

Type: Double

Start value: 0

Description: this sample example for creating variables

Global: false

Group: User's variables

Save Close

Имя переменной должно быть определено по имени java конвенций:

Нейминг

Каждый язык программирования имеет свой набор правил и соглашений для видов имен, которые разрешено использовать, и язык программирования java ничем не отличается. Правила и соглашения об именах переменных могут быть сведены к следующему:

Имена переменных чувствительны к регистру. Имя переменной может быть любой допустимый идентификатор - неограниченной длины последовательность Юникод букв и цифр, начинающийся с буквы, знака доллара "\$", или символ подчеркивания "_". Конвенция, однако, это всегда начинайте имена переменных с буквы, а не "\$" или "_". Кроме того, символ доллара, по Конвенции, не употребляется вовсе. Вы можете найти некоторые ситуации, в которых автоматически сгенерированные имена будут содержать знак доллара, но ваши имена переменных должны всегда избегать ее использования. Подобная Конвенция существует для подчеркивания; в то время как это технически законно начать переменную с именем "_", но эта практика не поощряется. Пробелы не допускаются.

Последующими символами могут быть буквы, цифры, знаки доллара, или символы подчеркивания. Конвенций (и здравого смысла), чтобы применить это правило. При выборе имен для переменных, используйте полные слова вместо загадочных аббревиатур. Это сделает ваш код проще для чтения и понимания. Во многих случаях это также сделает ваш код самодокументированным; поля по имени каденс, скорость, и шестерни, например, гораздо более интуитивны, чем сокращенные версии, такие как s, C и G. Также имейте в виду, что выбранное вами имя не должно быть ключевым или зарезервированным словом.

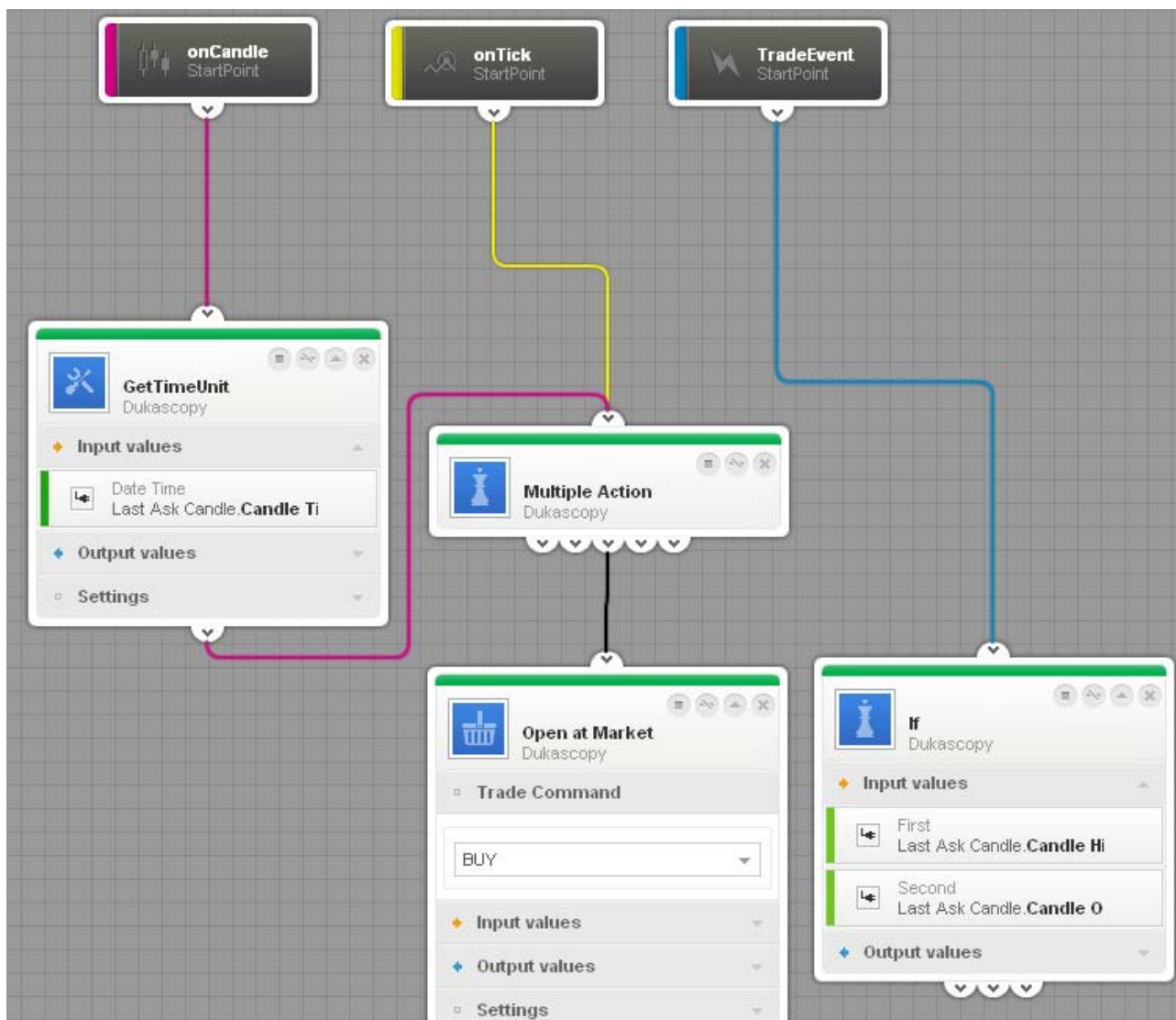
Если выбранное вами имя состоит из одного слова, пишите это слово в нижнем регистре. Если он состоит более чем из одного слова, сделайте заглавной первую букву каждого последующего слова. Имена `gearRatio` и `currentGear` являются яркими примерами настоящей Конвенции. Если переменная сохраняет постоянное значение, такие как статические заключительные инт `NUM_GEAR = 6`, Конвенция изменяется незначительно, спекулируя каждой буквы и отделять последующие слова с символа подчеркивания. По соглашению, символ подчеркивания никогда не используется в другом месте.

Начальная точка (StartPoint). Описание.

StartPoint (начальная точка) - это событие, которое генерируется системой. Каждое событие будет срабатывать (запускать) стратегию. Начальная точка может быть соединена с любым блоком и несколько начальных точек могут вызвать тот же блок. Невозможно вызвать начальную точку - только система создает событие. Существует три типа начальных точек:

- 1) **Trade Event** (торговое событие) - начнет свою стратегию на каждом новом торговом событии;
- 2) **onCandle** - начнет свою стратегию с каждой новой свечей;
- 3) **onTick** - начнет свою стратегию с каждым новым тиком.

Советы: нажав на левую вертикальные цветные линии блока, позволяет установить соответствующий цвет потока цвета. В случае, если начальные потоки смешиваются, то поток становится черным.



Описание типов данных (Data Type).

Целое Число (Integer).

Integer (Целочисленное) - это число, с разрядом данных 32-бит.

Она имеет минимальное значение и максимальное значение -2,147,483,648 2,147,483,647 (включительно).

Для интегральных значений, этот тип данных, как правило, по умолчанию выбор, если есть причина (как выше), чтобы выбрать что-то другое.

Этот тип данных, скорее всего, будет достаточно большим для чисел, которые ваша программа будет использовать, но, если вам нужен более широкий диапазон значений, используйте **Date and Time**.

С целочисленными вы можете сделать арифметические операции (см. блоки **Calculation** (вычисления) и **Calculation Expression** (вычислительных выражений)).



Дата и время (Date and Time) - тип данных

Date and Time - это тип данных, которое представляет собой 64-разрядное целое число.

Она имеет минимальное значение -9,223,372,036,854,775,808 и максимальным значением 9,223,372,036,854,775,807 (включительно).

Используйте этот тип данных, если необходим диапазон значений шире, чем те, которые предусмотрены целочисленным типом данных.



Тип Данных Double (Вещественный). Описание.

Double - это число, вещественный тип данных 64-разрядный с плавающей точкой.

С этим типом данных вы можете сделать арифметические операции (см. блоки **Calculation** (вычисления) и **Calculation Expression** (вычислительных выражений)).



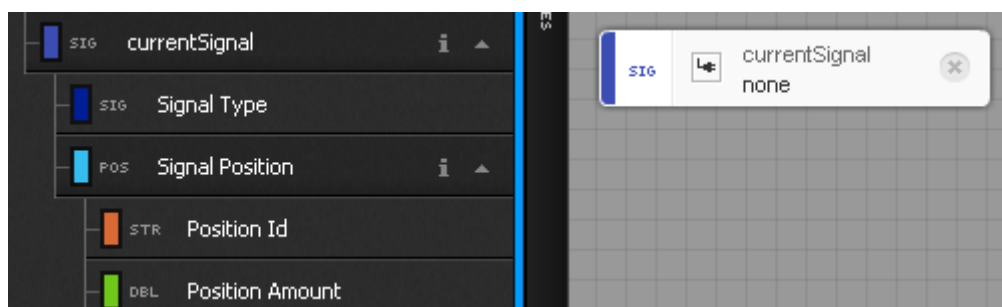
Тип данных Period (Период). Описание.

Period - тип переменной, которая представляет периоды баров/свечей. Этот период обычно используется при работе с данными свечи и с индикаторами. Он может принимать значение от 10 секунд до одного месяца.



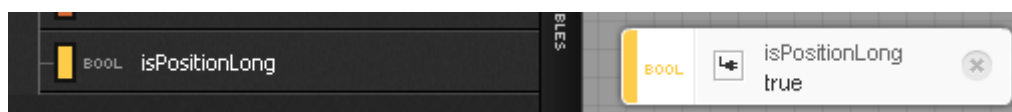
Тип данных Signal (Сигнал). Описание.

Signal - тип данных, который представляет собой торговый сигнал вывода, который генерирует стратегия, он используется в качестве компонента в базовой стратегии. Сигнал содержит информацию о сигнале типа и содержит сигнал установки в положение, или в информацию о ордере. Стратегия вывода мы используем массив сигналов, для чтения этого массива мы используем **Loop Viewer**, см. пример в стратегии.



Тип данных Boolean (Логический). Описание.

Boolean - логическое значение, значения которого может быть только: True (правда) или false (ложь), с другими словами, это логический тип переменной.



Тип данных String (Строковый). Описание.

String (Строка) – этот тип переменной, которая позволяет хранить текст. Вы можете хранить только один символ или множество символов.



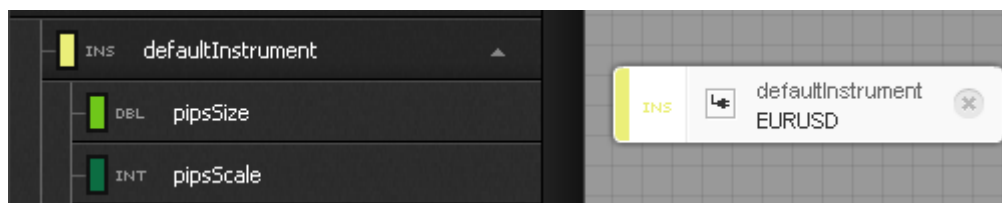
Тип данных Instrument (инструмент). Описание.

Instrument - этот тип переменной позволяет получить информацию об инструменте (валютной паре). Этот тип также содержит дополнительные методы:

1) **pipsSize** - Double значение типа, возвращает размер документа пипсов, например, для EURUSD он будет 0.0001

2) **pipsScale** - Целочисленное значение типа, возвращает пунктов шкалы, например, для EURUSD это будет 4.

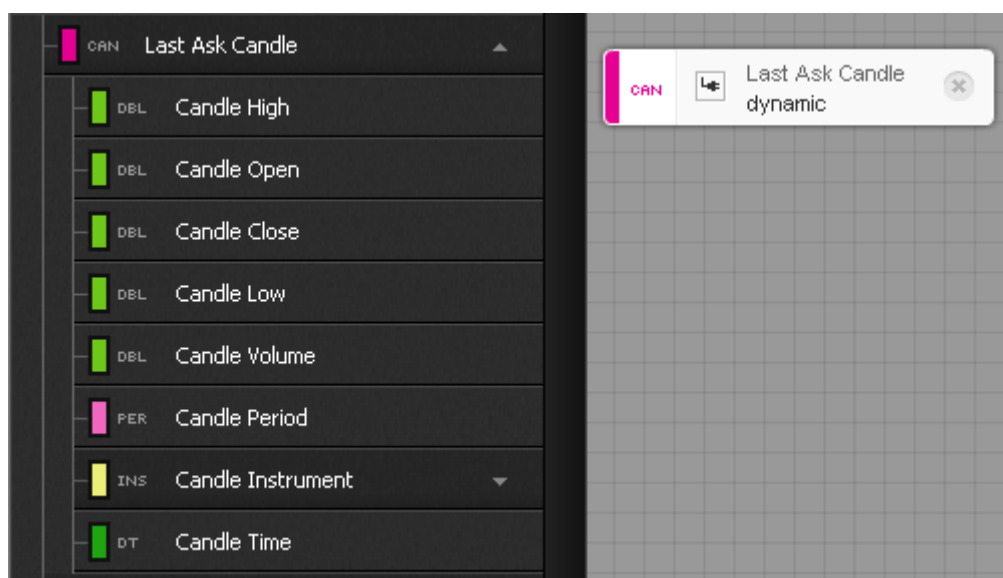
Этих методов возвращает системные переменные, и эти переменные величины могут быть изменены в будущем.



Тип данных Candle (Свеча). Описание.

Candle (Свеча) - этот тип переменной позволяет получить сведения о свече (баре). Этот тип также содержит дополнительные методы:

- 1) **Candle High** (Верх свечи) значение Double этого типа, возвращает максимальное значение свечи;
- 2) **Candle Open** (Открытие свечи) значение Double этого типа, возвращает значение цены открытия свечи;
- 3) **Candle Close** (Заккрытие свечи) значение Double этого типа, возвращает значение цены закрытия свечи;
- 4) **Candle Low** (низ свечи) значение Double этого типа, возвращает минимальное значение свечи.
- 5) **Candle Volume** (объем свечи) - значение Double этого типа, возвращает объем бара. Это сумма объемов лучших цен для каждого тика в этом баре;
- 6) **Candle Period** (период свечи) - значения периода, возвращает значение периода свечи;
- 7) **Candle Instrument** (инструмент свечи) - значение инструмента, возвращает значение инструмента свечи;
- 8) **Candle Time** (время свечи) - дата и время значение типа, возвращает время свечи.



Тип данных Messagetype (Массив). Описание.

Messagetype — этот тип переменной позволяет получить тип сообщения. Это может потребоваться, когда требуется найти конкретное сообщение о событии.



Тип данных OfferSide. Описание.

OfferSide - этот тип переменной определяет сторону предложения ask или bid.



Тип данных Command (команда). Описание.

Command - этот тип переменной указывает команду позицию или отложенного ордера.



Тип данных State (состояние). Описание.

State (состояние) - это тип переменной определяет состояние позиции или отложенного ордера (отменить, закрыть, создать, заполнить, открыть).



Тип данных AppliedPrice (приложение цены). Описание.

AppliedPrice - этот тип переменной указывает, какие цены использовать для индикатора входных данных для расчета.



Тип данных SignalType (Сигнал). Описание.

SignalType - этот тип переменной определяет, какой сигнал вырабатывает компонент стратегии.



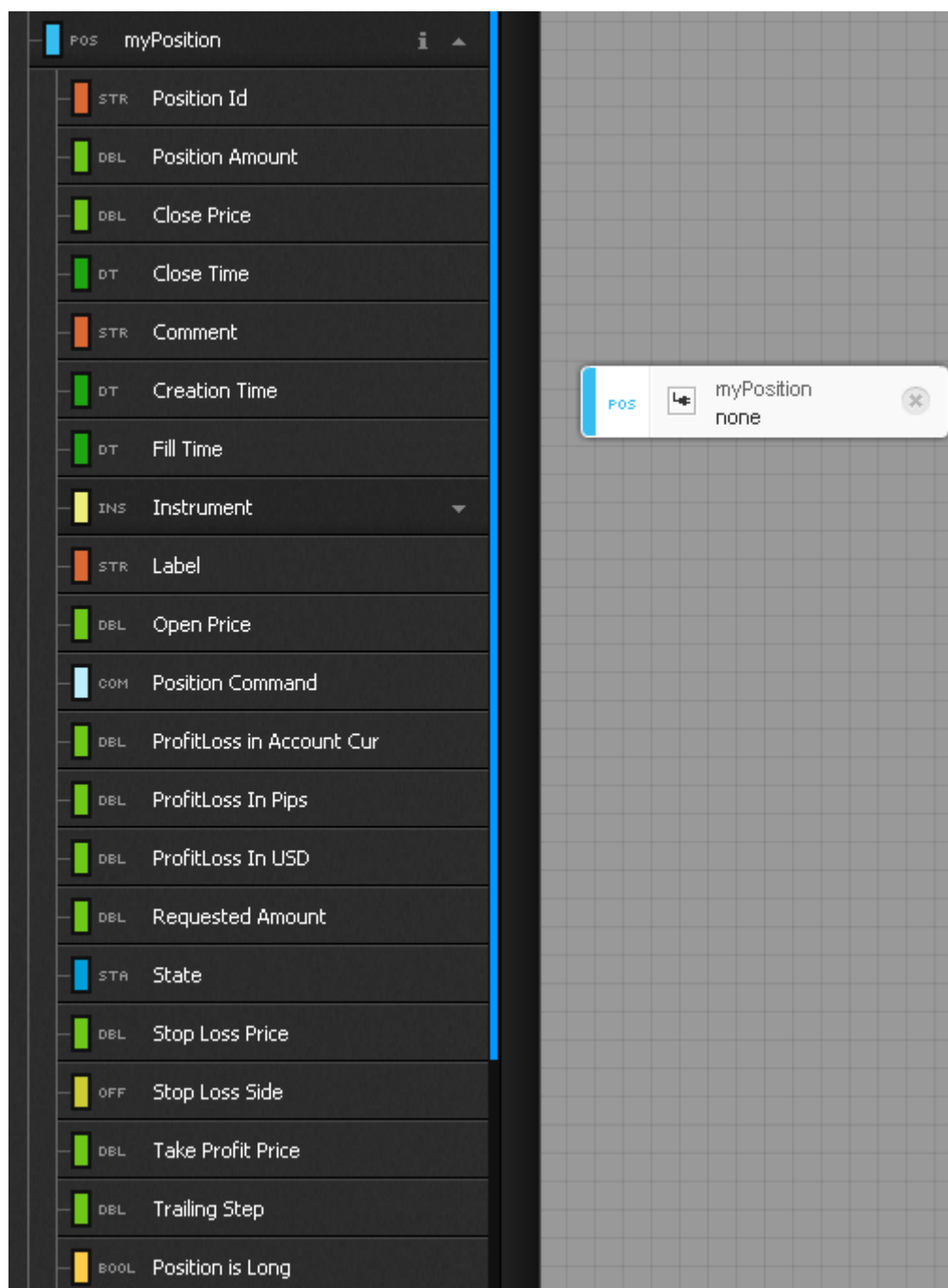
Тип данных MAType (Moving Average / Скользящее среднее). Описание.

MAType - это тип переменной определяет тип скользящих средних, который используется в индикаторе.



Тип данных Position (позиция). Описание.

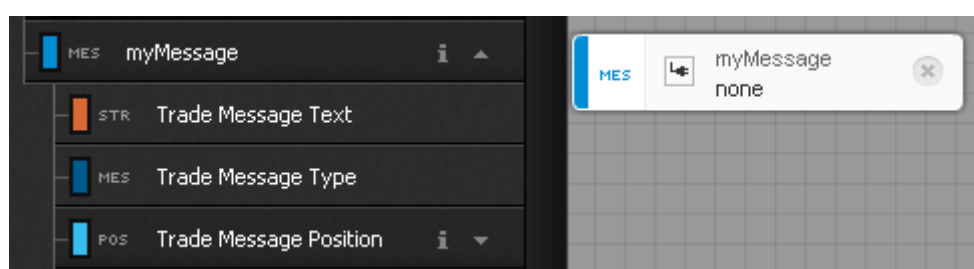
Position – этот тип переменной содержит данные о позиции/ордере. Это сложный тип переменной и содержит следующие данные/методы:



Тип данных Message (сообщение). Описание.

Message - это тип переменной представляет собой сообщение, отправленное с сервера на клиентское приложение. Это вывод из начальной точки (Start Point) - Trade Event. Это сообщение имеет следующие данные:

- 1) **Trade Message Text** текст торгового сообщения - строка, текстовой тип;
- 2) **Trade Message Type** тип торгового сообщения - значение класса messageType;
- 3) **Trade Message Position** позиция торгового сообщения - позиция типа.



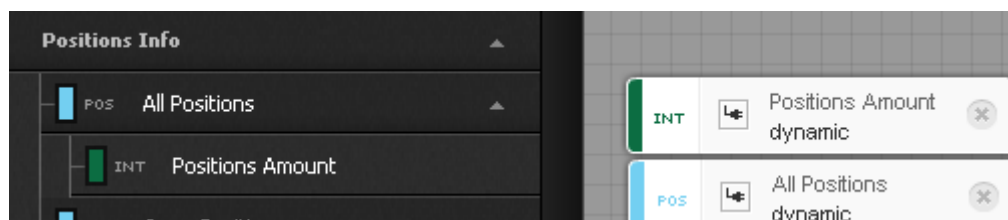
Тип данных Tick (Тик). Описание.

Tick - этот тип переменной позволяет получить информацию о тике. Тик это результат начальной точки (StartPoint) Tick. Этот тип также содержит следующие данные/методы:

- 1) **Tick Bid** (Bid тика) - Double значение типа, возвращает значение цены тика Bid;
- 2) **Tick Ask** (ASK тика) - Double значение типа, возвращает значение цены тика Ask;
- 3) **Tick Bid Volume** (уровень тика Bid) - Double значение типа, возвращает значение уровня Bid тика;
- 4) **Tick Ask Volume** (уровень тика Ask) - Double значение типа, возвращает значение уровня Ask тика;
- 5) **Tick Instrument** (инструмент тика) - значение типа инструмент (Instrument), возвращает значение инструмента тика;
- 6) **Tick Time** (время тика) - значение типа Дата и время (Date and Time), возвращает время тика.

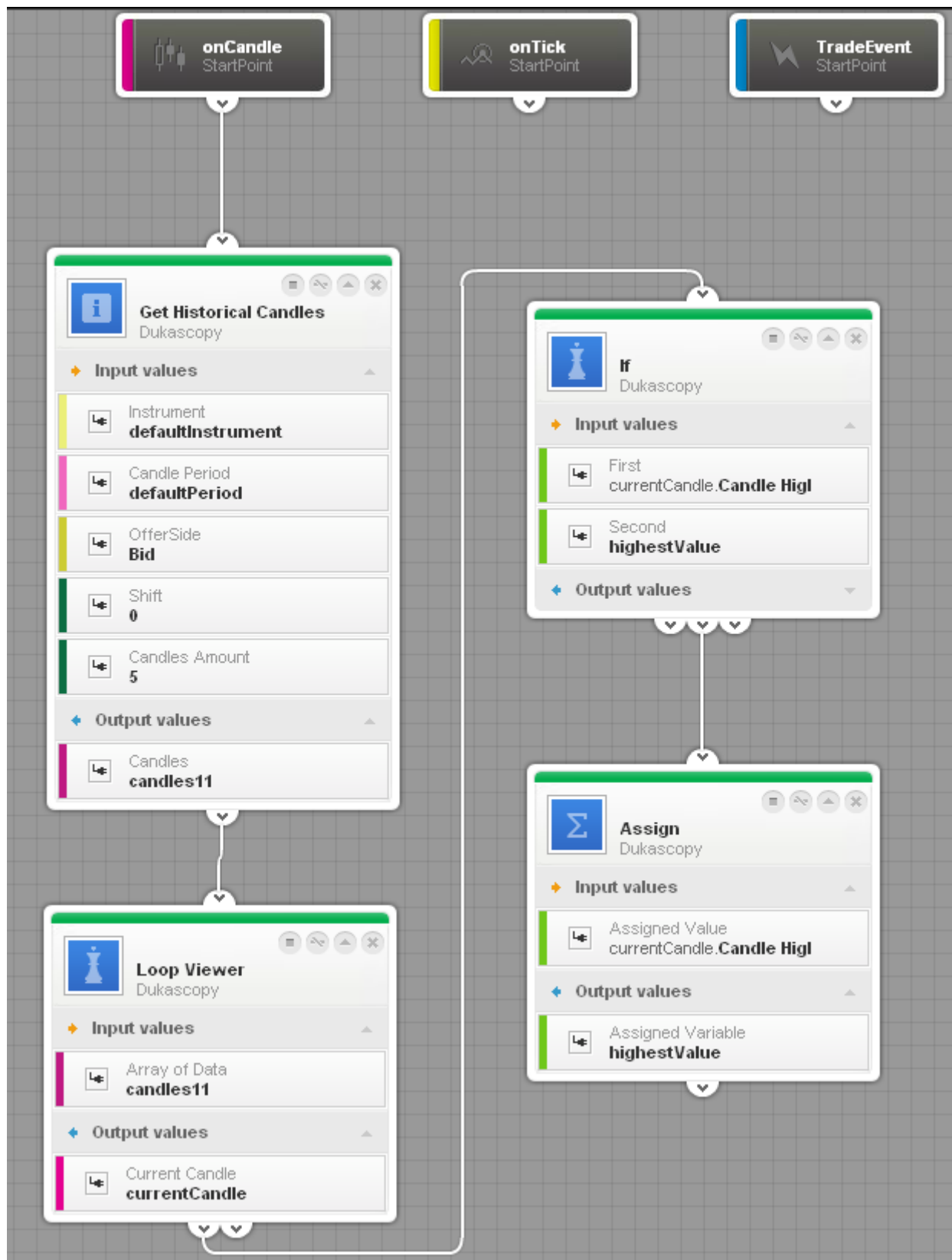
Тип данных Positions (Позиции). Описание.

Positions - этот тип переменной содержит данные о позиции/ордере или массиве ордеров. Для работы с массивами ордеров/позиций могут быть использованы **Positions Viewer** и **Loop Viewer**. Массив позиций имеет значение объема позиции, целочисленное значение.



Тип данных Candles (свечи). Описание.

Candles - этот тип переменных содержит тип свечи/бара или их массивов. Для работы с массивом позиции можно использовать **Loop Viewer**. Массив свечей имеет количество свечей, целочисленное значение. В следующих примерах показано, как получить массив исторических свечей и как найти наибольшее значение из этой коллекции:



Тип данных Signals (Сигналы). Описание.

Signals - это тип переменной, которая содержит массив сигналов или коллекции. Для работы с массивом позиции используйте **Loop Viewer**. См. пример **Loop Viewer**. Сбор сигналов имеет количество свечей, целочисленные данные.

