DtBuySellPressure 1.0.5.1+ API

Indicator factory method signature (build 1.0.5.1...3):

(see example of creation of indicator instance below)

```
NinjaTrader.NinjaScript.Indicators.DtBuySellPressure DtBuySellPressure(
	DiscoTrading.Nt8.DtBuySellPressure.Enums.DataSourceType dataSource,
	DiscoTrading.Nt8.DtBuySellPressure.Enums.VisualMode visualizationMode,
	DiscoTrading.Nt8.DtBuySellPressure.Enums.BaseMetric baseMetric,
	DiscoTrading.Nt8.DtBuySellPressure.Enums.DeltaMode deltaMode,
	bool cumulativeMode,
	bool resetOnSessionStart,

	DiscoTrading.Nt8.DtBuySellPressure.Enums.BracketMode bracketMode,
	long bracketThresholdVolume,

	DiscoTrading.Nt8.DtBuySellPressure.Enums.FilterModes filterMode,
	long minSize,
	long maxSize,
	string rowOfSizes,

	NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.DataTableAppearance
dataTableAppearance,
	NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.DeltaCandlestickChartAppearance
deltaCandlestickChartAppearance,
	NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.TotalVsDeltaBarChartAppearance
totalVsDeltaBarChartAppearance,
	NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.DeltaMarksAppearance
deltaMarksAppearance,
	NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.PocMarksAppearance
pocMarksAppearance,
	NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.DeltaConvDivMarksSettings
deltaConvDivMarksSettings
);
```

The first 12 input parameters (from ***visualizationMode...***to ***rowOfSizes***) are core parameters whose values have an impact on indicator's computations and output series/value therefore.
All other parameters are complex appearance properties which may be completely unimportant for use of the *DtBuySellPressure* indicator in custom NinjaScripts (custom indicators or strategies that only need to obtain outputs provided by *DtBuySellPressure* for internal further processing). I.e. you may set default values of those appearance parameters without any aftereffects on output values.
For example, simply use the default constructor to instantiate a DataTableAppearance properties object with default settings values:

```
new NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.DataTableAppearance();
```

Values of custom types:

Enumerations (possible values of enum input parameters)**:**

dataSource parameter:

```
DiscoTrading.Nt8.DtBuySellPressure.Enums.DataSourceType {
	TickReplay,
	TickSeries
}
```

visualizationMode parameter:

```
DiscoTrading.Nt8.DtBuySellPressure.Enums.VisualMode
{
	SubPanelDeltaCandlestickChart,
	SubPanelTotalVsDeltaBarChart,
	PriceChartDeltaAndPocMarks,
	DeltaConvDivMarkers,
}
```

baseMetric parameter:

```
DiscoTrading.Nt8.DtBuySellPressure.Enums.BaseMetric
{
	Volume,
	Ticks
}
```

deltaMode parameter:

```
DiscoTrading.Nt8.DtBuySellPressure.Enums.DeltaMode
{
	BidVsAsk,
	UpTickVsDownTick
}
```

bracketMode parameter:

```
DiscoTrading.Nt8.DtBuySellPressure.Enums.BracketMode
{
	Off,
	On
}
```

filterMode parameter:

```
DiscoTrading.Nt8.DtBuySellPressure.Enums.FilterModes
{
	NoFiltering,
	GreaterThanOrEqual,
	LessThanOrEqual,
	InBetween,
	RowOfSizes
}
```

barsMode parameter:

```
NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.BarsMode
{
	DeltaCandles,
	TotalWithDeltaBars
}
```

positiveDeltaArrangement and negativeDeltaArrangement parameters:

```
NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.VertialArrangement
{
	Above,
	Below
}
```

Type DiscoTrading.Nt8.DtBuySellPressure.PropertyTypes.*LineSettings*
(used in appearance property objects)

Constructor:
```
DiscoTrading.Nt8.DtBuySellPressure.PropertyTypes.LineSettings(Color color,
SharpDX.Direct2D1.DashStyle dashStyle, float width);
```

Default constructor LineSettings() creates the LineSettings instance with the following default property values:
```
Color = Colors.Blue;
DashStyle = SharpDX.Direct2D1.DashStyle.Solid;
Width = 1f;
```

Output Series (Type ***Name***)

- Series<Double> ***Delta***
- Series<double> ***DeltaHigh***
- Series<double> ***DeltaLow***

— these data series provide *Volume-based* or *Tick-based* Delta values depending on chosen ***baseMetric*** input parameter (BaseMetric.Volume or BaseMetric.Ticks).

- Series<long> ***Volume*** — Volume at the bar.

- Series<long> ***Ticks*** — Ticks (number of trades) at the bar.

Volume-based Delta series:

- Series<long> ***VolumeDeltaOpen***
- Series<long> ***VolumeDeltaHigh***
- Series<long> ***VolumeDeltaLow***
- Series<long> ***VolumeDeltaClose***

Tick-based Delta series:

Series<long> ***TickDeltaOpen***
Series<long> ***TickDeltaHigh***
Series<long> ***TickDeltaLow***
Series<long> ***TickDeltaClose***

Examples and best practice

Values of the output series can be accessed within the ***OnBarUpdate()*** method of your wrapper indicator.

Example of obtaining of current Volume-Delta and Tick-Delta values on the current bar:

```
protected override void OnBarUpdate()
{
	if (BarsInProgress != 0)
		return;

	long volumeDelta = 0;
	long tickDelta = 0;

	if (_buySellPressure.VolumeDeltaClose.IsValidDataPoint(0))
		volumeDelta = _buySellPressure.VolumeDeltaClose[0];
```

```
	if (_buySellPressure.TickDeltaClose.IsValidDataPoint(0))
		tickDelta = _buySellPressure.TickDeltaClose[0];

	...
	...
}
```

where **`_buySellPressure`** is a custom indicator's global variable that holds an instance of **`NinjaTrader.NinjaScript.Indicators.`*`DtBuySellPressure`*** type.

An instance of the ***DtBuySellPressure*** must be created in the ***OnStateChange()*** method when the **State** property == **State.Configure**. Example:

```
using DiscoTrading.Nt8.DtBuySellPressure.Enums;

public class YourCustomIndicator: Indicator
{
	// declare global variable to hold an instance of the DtBuySellPressure
	private NinjaTrader.NinjaScript.Indicators.DtBuySellPressure _buySellPressure = null;

	protected override void OnStateChange()
	{
		if (State == State.Configure)
		{
			// create an instance of the DtBuySellPressure indicator which can be
			// used to obtain Cumulative Delta values.
			// This instance will utilize TickSeries as a DataSource, use Volume as
			// a base metric and “Bid Vs Ask” technique of Delta calculation
			_buySellPressure = NinjaTrader.NinjaScript.Indicators.DtBuySellPressure(
				DataSourceType.TickSeries,
				VisualMode.SubPanelDeltaCandlestickChart,
				BaseMetric.Volume,
				DeltaMode.BidVsAsk,
				true,
				true,
				BracketMode.Off,
				100000,
				FilterModes.NoFiltering,
				0,
				99999,
				string.Empty,
				new DtBuySellPressure.DataTableAppearance(),
				new DtBuySellPressure.DeltaCandlestickChartAppearance(),
				new DtBuySellPressure.TotalVsDeltaBarChartAppearance(),
				new DtBuySellPressure.DeltaMarksAppearance(),
				new DtBuySellPressure.PocMarksAppearance(),
				new DtBuySellPressure.DeltaConvDivMarksSettings()
			);

		}
	}
	// other methods...
	...
}
```