

DtBuySellPressure 1.0.4.7 API

Indicator factory method signature (build 1.0.4.7):

(see example of creation of indicator instance below)

```
NinjaTrader.NinjaScript.Indicators.DtBuySellPressure DtBuySellPressure(  
NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.VisualMode visualizationMode,  
NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.DataSourceType dataSource,  
NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.BaseMetric baseMetric,  
NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.DeltaMode deltaMode,  
bool cumulativeMode,  
bool resetOnSessionStart,  
NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.BasketMode bracketMode,  
long bracketThresholdVolume,  
NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.FilterModes filterMode,  
long minSize,  
long maxSize,  
string rowOfSizes,  
  
bool showSecondaryMetricRows,  
bool showTotalValues,  
bool showDeltaValues,  
bool showMaxMinValues,  
bool showMinusSign,  
NinjaTrader.Gui.Tools.SimpleFont numbersFont,  
Color numbersNeutralTextColor,  
Color numbersNeutralBackColor,  
Color numbersPositiveTextColor,  
Color numbersPositiveBackColor,  
Color numbersNegativeTextColor,  
Color numbersNegativeBackColor,  
bool numbersFadeColors,  
Color numbersPositiveBackColorMax,  
Color numbersNegativeBackColorMax,  
  
NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.BarsMode barsMode,  
bool plotWick,  
float wickWidth,  
bool fixedBarWidth,  
float barWidth,  
Color colorTotalBar,  
Color colorPositiveDelta,  
Color colorNegativeDelta,  
Color colorMaximumDelta,  
Color colorMinimumDelta,  
Color colorNeutralDelta,  
DiscoTrading.Nt8.DtBuySellPressure.PropertyTypes.LineSettings outlinePositiveDelta,  
DiscoTrading.Nt8.DtBuySellPressure.PropertyTypes.LineSettings outlineNegativeDelta,  
DiscoTrading.Nt8.DtBuySellPressure.PropertyTypes.LineSettings zeroLine,  
  
bool showSecondaryMetric,  
NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.VerticalArrangement positiveDeltaArrangement,  
NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.VerticalArrangement negativeDeltaArrangement,  
int spacing,  
int labelWidth,
```

```

bool labelOutline,
int ratioBarHeight,
int numOfBarsToPlot,
bool showAggregateMarker,
bool showDeltaOfCurrentBar,
int currentBarMarkShift,

bool showVpocMarks,
Color vpocMarksColor,
DiscoTrading.Nt8.DtBuySellPressure.PropertyTypes.LineSettings vpocMarksOutline,
bool vpocMarksAutoSize,
float vpocMarksWidth,
float vpocMarksHeight
);

```

The first 12 input parameters (from **visualizationMode...** to **rowOfSizes**) are core parameters which values have an impact on indicator's computations and output series/value therefore.

All other parameters are appearance options 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.

Values of custom types:

Enumerations (possible values of enum input parameters):

visualizationMode parameter:

```

NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.VisualMode
{
    SubPanelDeltaCandlestickChart,
    PriceChartDeltaPercentBars,
    DeltaConvDivMarkers,
}

```

dataSource parameter:

```

NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.DtBuySellPressure.DataSourceType {
    TickReplay,
    TickSeries
}

```

baseMetric parameter:

```

NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.DtBuySellPressure.BaseMetric
{
    Volume,
    Ticks
}

```

deltaMode parameter:

```

NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.DtBuySellPressure.DeltaMode
{
    BidVsAsk,
    UpTickVsDownTick
}

```

bracketMode parameter:

```
NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.BracketMode
{
    Off,
    On
}
```

filterMode parameter:

```
NinjaTrader.NinjaScript.Indicators.DtBuySellPressure.FilterModes
{
    NoFiltering,
    GreaterThanOrEqual,
    LessThanOrEqual,
    InBetween,
    RowOfSizes
}
```

barsMode parameter:

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

positiveDeltaArrangement and negativeDeltaArrangement parameters:

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

Type DiscoTrading.Nt8.DtBuySellPressure.PropertyTypes.LineSettings

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 properties:

```
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];

    ...
    ...
}
```

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

```
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(
                Indicators.DtBuySellPressure.VisualMode.SubPanelDeltaCandlestickChart,
                Indicators.DtBuySellPressure.DataSourceType.TickSeries,
                Indicators.DtBuySellPressure.BaseMetric.Volume,
```

```

Indicators.DtBuySellPressure.DeltaMode.BidVsAsk,
true,
true,
Indicators.DtBuySellPressure.BasketMode.Off,
100000,
Indicators.DtBuySellPressure.FilterModes.NoFiltering,
0,
99999,
string.Empty,
false,
false,
false,
false,
false,
new SimpleFont(),
Colors.Black,
Colors.Black,
Colors.Black,
Colors.Green,
Colors.Black,
Colors.DarkRed,
false,
Colors.Black,
Colors.Black,
Indicators.DtBuySellPressure.BarsMode.DeltaCandles,
true,
2f,
false,
10f,
Colors.DimGray,
Colors.Green,
Colors.DarkRed,
Colors.Black,
Colors.Black,
Colors.Black,
new LineSettings(),
new LineSettings(),
new LineSettings(),
false,
Indicators.DtBuySellPressure.VerticalArrangement.Below,
Indicators.DtBuySellPressure.VerticalArrangement.Above,
10,
0,
false,
5,
0,
false,
false,
0,
false,
Colors.Blue,
new LineSettings(),
true,
10,
10
);

}

// other methods...

```

