MZpack 3.18.1
MZpack API 2.2.11 for NinjaTrader 8
Rev 1.9

USER GUIDE

WWW.MZPACK.PRO

CONTENT

Contacts, Links, and Social

- Knowledge Base https://mzpack.freshdesk.com/en/support/solutions/
- Helpdesk https://mzpack.freshdesk.com/en/support/home
- WWW https://www.mzpack.pro
- MZpack Trading Room in Skype https://join.skype.com/B48OjcjNeodD
- MZpack YouTube channel http://www.youtube.com/channel/UCqabMIgUEjk3Bkee8G-oSMg
- Facebook https://www.facebook.com/mzpackninjatraderindicators/
- Newsletters subscription and feedback form http://www.mzpack.pro/en/contact-mzpack-ninjatrader-indicators-team/

Introduction

On what information and data the MZpack indicators for NinjaTrader are based

- Tick-by-tick trades
- Proprietary Market Microstructure Events set
- Level I & Level II
- CME MDP 3.0

MZpack trading tools can be used in variety of applications from scalping to day trading. Also momentum and swing traders can use MZpack to spot entries and manage trades/exits.

Benefits of MZpack indicators

- See inside the price action and inside the bars.
- Unique **Market Microstructure patterns** discovered and developed
- Deep cross volume-based: both horizontal and vertical
- Flat ranges/brackets analysis to spot future momentum side
- Real-time with no lagging and historical too

MZpack 3 for NinjaTrader 8 works on historical data with **Tick Replay** option checked. But some of futures like Iceberg orders and Smart/Predatory trades searching and mzMarketDepth indicator work in real-time only because NinjaTrader 8 does not provide Level I & Level II on historical data.

Installing MZpack Indicators for NinjaTrader 8

1. Download MZpack Indicators installer from the site. **We recommend 64-bit installer for Windows 64-bit users.**

2. Close NinjaTrader 8 if it was opened. Run MZpack Indicators .exe installer.

3. After MZpack Indicators installation finished run **NinjaTrader 8**. NinjaTrader **8.0.25.0** version or later required.

4. Make sure Show Tick replay option enabled (menu Tools – Options – Market data - Show Tick Replay)
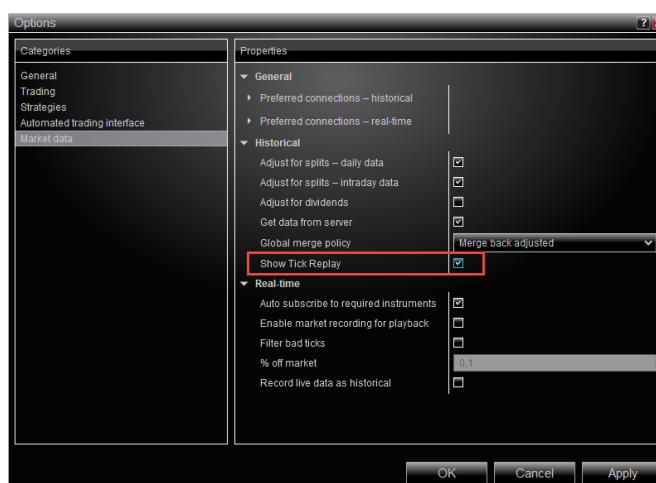
5. Open a chart. Click right mouse button and click Data Series from the chart context menu. Turn on Tick Replay option for historical Bid/Ask data support. Note, mzVolumeProfile indicator in some modes doesn't require Tick Rplay.
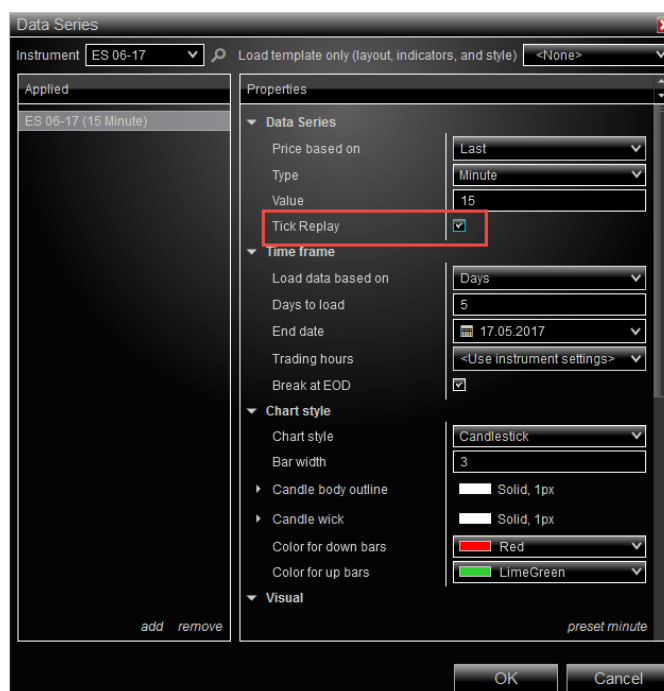
6. Open Indicators window (Ctrl+I) and add an MZpack indicator. Click Ok.

7. Activating window will be shown. If Activating window is not visible try to find it on Windows Task bar. Enter license key and click **Activate** button. Activate button is enabled in paid version only. Click **Start Trial** button if you installed **Trial** version.
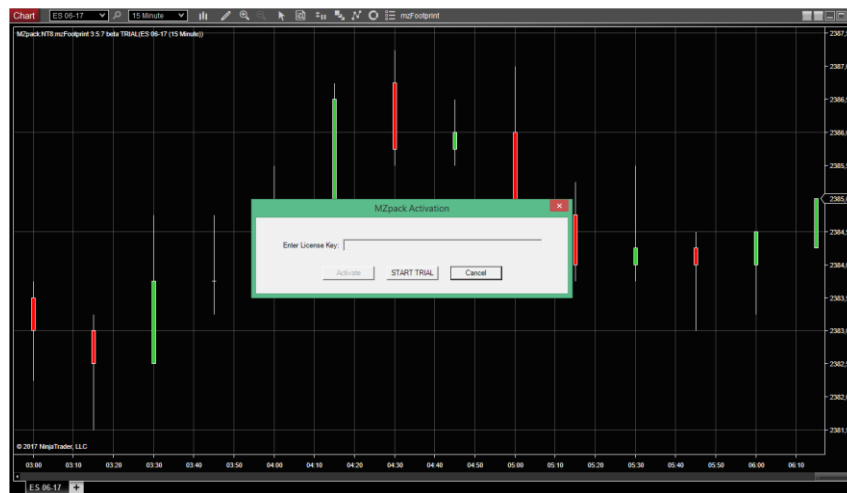
8. Click Ok button in Activating message window, **restart NinjaTrader 8** and establish a data connection.

 How to start MZpack Indicators Trial?
https://www.youtube.com/watch?v=_YsTAndd8xk

6

QuickStart. Indicators Setup, Chart Templates and Workspaces

MZpack installation puts set of templates in corresponding NinjaTrader 8 folders. Check available chart templates by clicking **Templates – Load** from chart context menu.
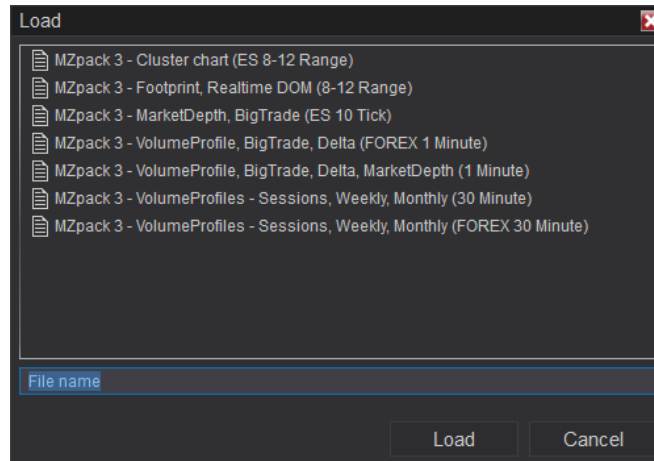
Workspaces are available from Workspaces main menu. Loading of some workspaces e.g. *MZpack 3 - VolumeProfiles - Sessions, Weekly, Monthly (30 Minute)* with Days to load = 30 might take some time (30-60 seconds depending on performance of your computer).

Check for workspaces updates here: https://www.mzpack.pro/ninjatrader-chart-templates-and-workspaces/
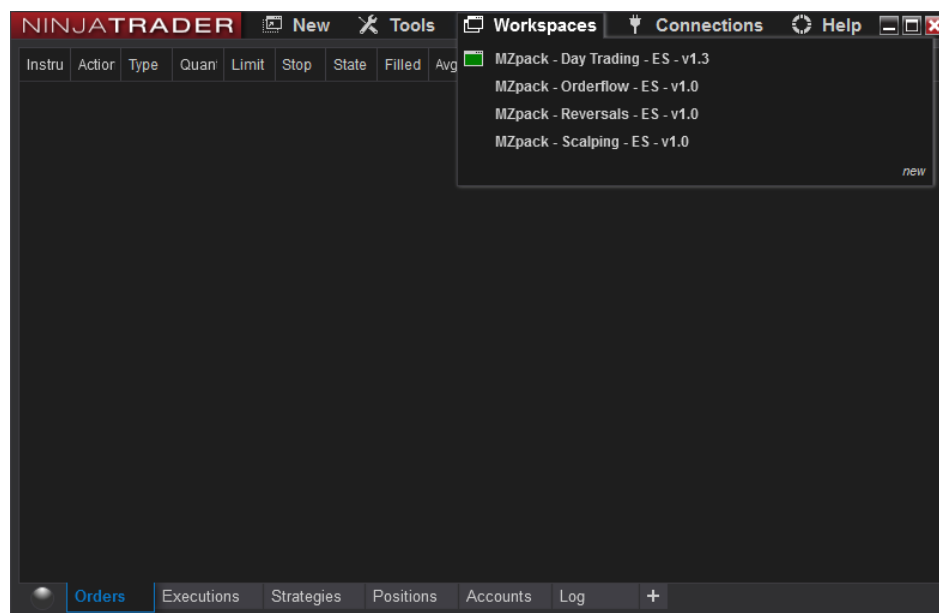
Performance Notes and Optimization

MZpack indicators are tick indicators processing market data on tick-by-tick basis.

There are two aspects of the optimization: for historical and live market data processing, and rendering also.

To calculate values on historical data with maximal precision we must use Tick replay mode. Tick replay uses historical data to build the Data Series. This requires additional PC resources adversary impacting the performance and initial calculation times of indicators and strategies.

First of all

- Check for products' updates. Update your MZpack software regularly.
- Read our newsletters and News on the site, please. We send newsletters to all customers and registered users

To reduce chart loading and initial calculation time:

- Set 'Days to load' to minimal possible value
- Remove unused indicators from the chart
- Close unused shadow workspaces

To optimize live market data processing and rendering, including avoiding chart delays:

- Remove unused indicators from the chart. If you need an indicator periodically only, turn its visibility on and off by clicking the 'eye' button <) on the tool bar next to its name.
- Close unused shadow workspaces
- Turn on 'General - Optimize render performance'. Set 'Maximal render time, ms' = 50..20. The chart might 'flash' after this but will be responsive.
- mzBigTrade: set 'Orderflow - Reconstruct tape apply' = ChartReload (this will save a lot of memory)
- mzMarketDepth: set 'Common - Depth of market' = 10..30. For some instruments/data subscriptions full order book is available.
- mzMarketDepth: set 'Historical DOM - History depth, bars' = 500..2000
- mzVolumeProfile, mzFootprint: set 'Ticks per level' = 2 or more
- mzVolumeProfile: set 'Volume profile - Ladders saturation' = OFF if 'Profile view' = Ladders
- mzVolumeProfile: turn TPO feature on and off if you need it periodically only
- mzVolumeProfile: set 'Volume profile - Profile accuracy' = Minute (for daily/weekly/etc. time frames and very large composite profiles)

Check Indicators' Settings topic in the Solution Center

Support of Non-Bid/Ask Historical Data: Forex, NSE, Cryptocurrencies, Stock Markets, etc.

If your market data provider doesn't support historical Bid-Ask data you will see all historical trades on the Bid side. Eg. **Forex** market doesn't provide historical Bid-Ask data.

These indicators can be adjusted for historical non-BidAsk data: mzFootprint, mzVolumeProfile, mzBigTrade, mzVolumeDelta. Choose 'Orderflow – Calculation mode' = **UpDownTick**.

For Forex pairs, we recommend using relevant futures to enable all features of MZpack in your analysis (including DOM analysis). Eg. for EURUSD use 6E futures.

You can get free limited CQG subscription here https://ninjatrader.com/FreeLiveData

ORDER FLOW HYBRID MODE AND NSE (INDIAN MARKET)

NSE Market data providers (eg. Accelpix) don't translate historical Bid-Ask data so far. For NSE users, **Hybrid** mode for *Orderflow – Calculation mode* settings has been added. The Hybrid mode uses **UpDownTick** calculation for historical data, and **BidAsk** calculation for real-time data. BidAsk attribution of trades is 100% accurate. Also, adjust **Non-Crypto volumes divider** and **Ticks per level** parameters. Read more.

https://www.youtube.com/watch?v=5u0Z5e5kbiU

mzFootprint Indicator

Video Guide (English) https://www.mzpack.pro/product/mzfootprint-for-ninjatrader-8/

Video Review (German) https://www.youtube.com/watch?v=DVDhrqTg9rM

Indicator page: https://www.mzpack.pro/product/mzfootprint-for-ninjatrader-8/

Indicator settings

Left Footprint. Left side Footprint settings

- **Footprint style**
  - a. **BidAsk**. Classical Bid x Ask Footprint
  - b. **Volume**. Traded volume
  - c. **Delta**. Bid-Ask volumes delta
  - d. **DeltaPercentage**. Delta % = Delta/Volume %
  - e. **TradesNumber**. Shows number of trades in a cluster.
  - **f. Bid**
  - **g. Ask**
  - **h. None**
- **Cluster style**
  - a. **Brick**. Solid color fill of clusters
  - b. **Histogram**. Partial solid color fill of clusters proportionally to their values
  - c. **None**. No color fill
- **Cluster scale**
  - a. Bar. Histogram scale calculated inside current bar only
  - b. Chart. Histogram scale calculated among all bars visible on chart
  - c. All. Histogram scale calculated among all loaded bars
- **Scale source**
  If Cluster style is Histogram
  - a. Volume. Histogram is scaled by cluster volume
  - b. Delta. Histogram is scaled by cluster delta
  - c. TradesNumber. Histogram is scaled by number of trades in a cluster
- **Color source**
  If Color mode is Solid or Saturation
  - a. Volume. Color from Cluster parameter is used

b. Delta. Colors from Negative/positive delta are used. E.g. if Scale source = Volume and Color source = Delta you can see volume histograms colored by delta of each cluster

c. TradesNumber. Color from Cluster parameter is used

- **Gradient source**

  If Color mode is Saturation, Heatmap, GrayScaleHeatmap or Custom

  a. Volume. Color opacity or gradient is set by clusters' volumes

  b. Delta. Color opacity or gradient is set by clusters' deltas

  c. TradesNumber. Color opacity or gradient is set by number of trades in clusters

- **Cluster**. Color for clusters

- **Negative delta**. Color for negative delta (including slot color for Brick/Histogram styles

- **Positive delta**. Color for positive delta (including slot color for Brick/Histogram styles

- **Cluster border**

- **Color mode**

  a. Solid. Solid color mode

  b. Saturation. Saturation depending on values

  c. Heatmap. Heatmap colors by values

  d. GrayScaleHeatmap. Gay scaled heatmap colors by values

  e. Custom. Custom filters and colors are used

- **Custom 'less' filter.** Values less than this value are rendered by **Custom color 'less'** color

- **Custom '>=' filter #1.** Values greater than or equal to this value are rendered by **Custom color #1** color

- **Custom '>=' filter #2.** Values greater than or equal to this value are rendered by **Custom color #2** color

- **Custom '>=' filter #3.** Values greater than or equal to this value are rendered by **Custom color #3** color

- **Values: show.** Show cluster value (bid x ask, volume, delta, trades number)

- **Values: align.** Inner or Center

- **Values: divider**. Divide values by this divider to fit them in clusters.

- **Values: decimal places**. Decimal places after period for values if Values divider is 10 or greater.

- **Values: abs Delta.** Disable to see signed Delta when Color source is Delta.

- **Values: color**. Color for value in a cluster

- **Values: font**.

- **Cluster Zones: enable.** Enable Cluster Zones which can be **LVN (**Low Volume Nodes), High Volume Nodes (**HVN**), Delta ranges, Delta Percentage ranges, Trades ranges. Values from **Left** clusters are used here.
- **Cluster Zones: filter min.** Filter by minimal cluster value.
- **Cluster Zones: filter max.** Filter by maximal cluster value. Use -1 for no limit.
- **Cluster Zones: break on session.** Break zones on new session
- **Cluster Zones: style.** Style is Line or Zone
- **Cluster Zones: color.**

Right Footprint. Right side Footprint settings

- See **Left Footprint** …

Filters

- **Ticks per level**. Sets the level of aggregation for individual price levels, if price levels should be merged together. *Ticks per level is not available if Orderflow – Reconstruct tape is enabled.*
- **Trade min**. Minimal single or reconstructed trade volume cumulated by Footprint.
- **Trade max**. Maximal single or reconstructed trade volume cumulated by Footprint.
- **Display value**. All values and its corresponding clusters less than this parameter will not be shown

Presentation

- **Bid**. Color for prints on Bid side.
- **Ask**. Color for prints on Ask side.
- **Auto-scale values**.
- **Bar border**. Show bar border
- **Bar border**. Bar borderline
- **Bar marker**. Turn it on to replace default chart view style by Open-Close bar markers
- **Bar space, px**. Normally, you set this value using chart UI
- **Bar width, px**. Normally, you set this value using chart UI
- Bar width, px. Normally, you set this value using chart UI
- **Bar outer margin, px**. Set interbar space.
- **Control right margin**. Control right chart margin automatically to fit Footprint.
- **Chart right margin, px**. Override Ninja's margin settings.

Bar Statistic

- **Volume**. Show bar volume above or under the bar

- **Delta**. Show bar delta above or under the bar
- **Absolute Delta Average**. Show bar Bid/Ask absolute delta average. Calculated as sum of all absolute Deltas for each price level of the bar
- **Min/Max Delta**.
- **Delta %**
- **COT.** Commitment of Traders
- **Ratio Numbers: enable.** Enable Ratio Numbers statistic
- **Ratio Numbers: bounds low.** See Ratio Numbers topic
- **Ratio Numbers: bounds high.** See Ratio Numbers topic
- **Ratio Numbers: NEUTRAL.** See Ratio Numbers topic
- **Ratio Numbers: REJECTED/DEFENDED.** See Ratio Numbers topic
- **Values divider.** Displayed values will be divided by this divider.
- **Bid Delta**. Bid or Sell Delta color
- **Ask Delta**. Ask or Buy Delta color
- **Font**.

Statistics Grid

- **Show.** Show statistics grid
- Show legend
- **Legend position**. Left or Right
- Trades
- Volume
- Buy volume
- Sell volume
- Delta
- **Delta %**. Delta % = Delta / Volume * 100
- **Absolute Delta Average**. Calculated as sum of all absolute Deltas for each price level of the bar
- Delta cumulative
- Min Delta
- Max Delta
- Delta change
- **Delta rate**. Delta Rate value is attached to the bar and is shown in Statistics Grid, but is calculated continuously. Only maximal (modulo) value of Delta Rate is shown for the bar
- **Delta rate: type**. Tick or Millisecond
- **Delta rate: value**. Value of Delta rate measure interval in ticks or milliseconds
- **Delta rate: show in bar.** Show Delta Rate in bars as a vertical line for the price range at which Delta rate occurred

- COT Low
- COT High
- **Predicted values: show**. Stat values are being extrapolated (proportional to the bar time). Available for time-based intraday bar types only.
- **Predicted values: gauge.** Show time gauge and bar timer.
- **Values divider.** Displayed values will be divided by this divider.
- **Cell height, px**
- **Cell color scale**. Opacity is used to show value for a statistics cell. Chart – scale values by using visible bars. All – scale values by using all loaded data.
- **Cell border**. Show cell border
- **Cell border**. Line - contour for cells
- **Auto-scale values**. Auto-scale values in the grid.
- **Font**. Values font
- **Auto-scale bars**. Fit all chart bars above statistics grid.
- **Values color**.

Imbalance

- **Show**. Show diagonal Bid-Ask imbalance
- **Only Imbalance.** Show only imbalances
- **Imbalance, %.** Imbalances percentage
- **Filter**. Filter by traded volume on imbalance side
- **Marker: visibility**. None – not visible, NoValues – visible only when values are to small to be rendered, Always – marker is always visible
- **Marker: type.** Dot or Cluster
- **Marker: position.** Inner, Center, Outer
- **Sell/Resistance zone**. Color for sell imbalance marker/Resistance zone
- **Buy/Support zone**. Color for buy imbalance marker/Support zone
- **Font**
- **S/R zones: enable.** Show Support/Resistance zones based on levels of imbalances
- **S/R zones: consecutive levels**. Minimal number of consecutive levels in a zone.
- **S/R zones: volume filter**. Filter by total volume of S/R zone
- **S/R zones: ended by**. ByBarHighLow - zone ended when high or low of the bar (depending on support or resistance) crosses zone; ByBarClose - zone ended when close of the bar crosses zone; ByBarPOC - zone ended when POC of the bar crosses zone
- **S/R zones: break on session.** Break zones on new session
- **S/R zones: opacity, %.** Opacity for zone color

- **S/R zones: alert**. Use alert when a zone will be detected
- **S/R zones: support zone sound.** .wav file for support zone alert.
- **S/R zones: resistance zone sound nd.** .wav file for resistance zone alert.

Absorption.

- **Show**
- **Only Absorption**. Show only absorptions
- **Absorption, %.** Absorptions percentage
- **Depth**. Absorption depth means how far the price bounces from the level of absorption. In ticks
- **Filter**. Filter by traded volume on absorption side
- **Sell/Support zone**. Color for sell absorption marker/Support zone
- **Buy/Resistance zone**. Color for buy absorption marker/Resistance zone
- **S/R zones: enable.** Show Support/Resistance zones based on levels of imbalances
- **S/R zones: consecutive levels**. Minimal number of consecutive levels in a zone.
- **S/R zones: volume filter**. Filter by total volume of S/R zone
- **S/R zones: ended by**. ByBarHighLow - zone ended when high or low of the bar (depending on support or resistance) crosses zone; ByBarClose - zone ended when close of the bar crosses zone; ByBarPOC - zone ended when POC of the bar crosses zone
- **S/R zones: break on session.** Break zones on new session
- **S/R zones: opacity, %.** Opacity for zone color
- **S/R zones: alert**. Use alert when a zone will be detected
- **S/R zones: support zone sound.** .wav file for support zone alert.

Unfinished Auction

- Show
- Color. Color for Unfinished Auction cluster
- Opacity, %.
- Border. Color of border for Unfinished Auction cluster

Bar Volume Profile

- **POC**. Show Point Of Control for each bar
- **POC**. POC line
- **VA**. Show bar Value Area
- **VA, %**. Value Area percentage
- **VA**. Value Area color

- **VA opacity, %.**

Session/Daily Volume Profile

- Profile mode. Profile mode for volume-based levels: Session, Daily. Use Daily for Forex market.
- POC. Show developing Point Of Control for each session
- POC. Session POC line
- VA. Show session developed Value Area
- VA, %. Session Value Area percentage
- VA. Session Value Area line

Notifications

- **Trades number: alert.** Enable alert on trades number in a bar
- **Trades number: threshold.**
- **Trades number: sound.**
- **Volume: alert.** Enable alert on volume in a bar
- **Volume: threshold.**
- **Volume: sound.**
- **Buy volume: alert.** Enable alert on buy volume in a bar
- **Buy volume: threshold.**
- **Buy volume: sound.**
- **Sell volume: alert.** Enable alert on sell volume in a bar
- **Sell volume: threshold.**
- **Sell volume: sound.**
- **Delta: alert.** Enable alert on delta in a bar
- **Delta: threshold.**
- **Delta: sound.**
- …
- **Left cluster: alert.** Enable alert on value in a left cluster
- **Left cluster: threshold.**
- **Left cluster: sound.**
- **Right cluster: alert.** Enable alert on value in a right cluster
- **Right cluster: threshold.**
- **Right cluster: sound.**
- **Imbalance: alert**
- **Imbalance: sound**
- **Absorption: alert**
- **Absorption: sound**

- **Imbalance/Absorption: send Email.** Send email message on Imbalance/Absorption
- **Imbalance/Absorption: email address**

This indicator supports OnBarClose mode for optimization of system resources.
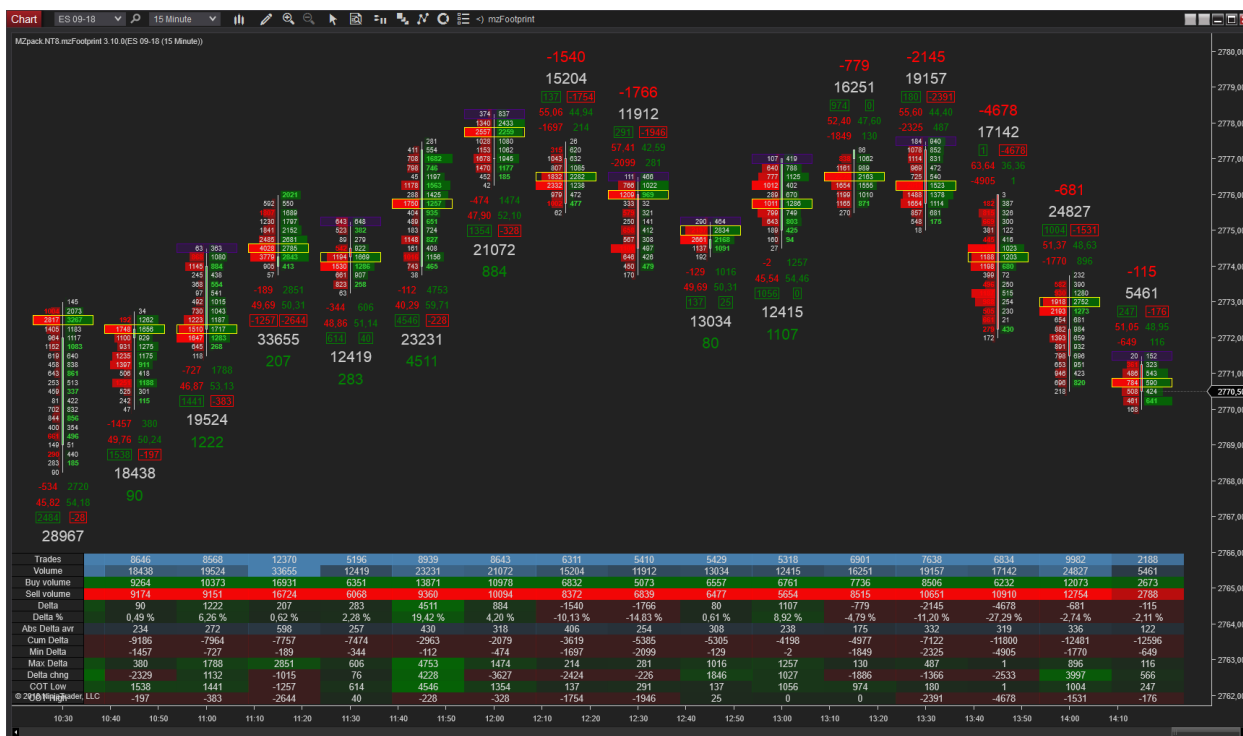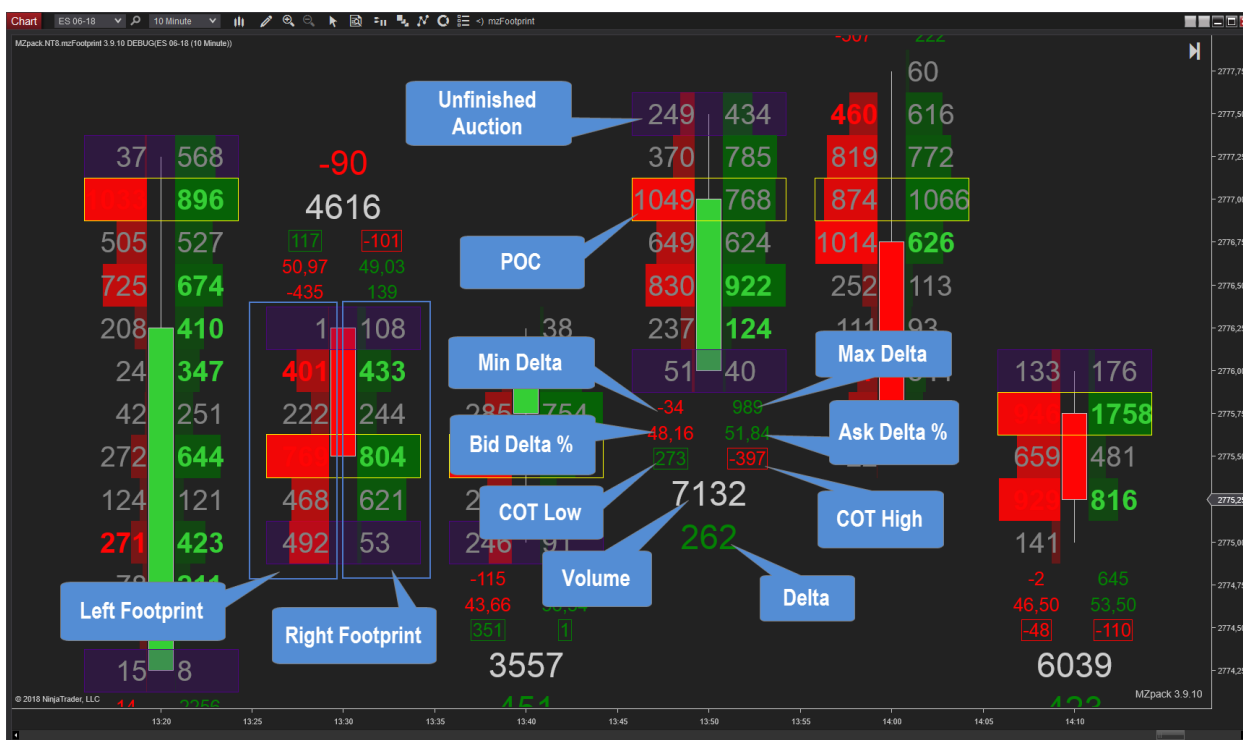
EXPLAINING IMBALANCE/ABSORPTION

mzFootprint calculates diagonal Imbalance. Diagonal imbalance e.g. at Ask means that volume of filled Buy orders is greater by given percentage than volume of filled Sell orders at price just below that Ask price. See fig. 71-Lot at Ask @2384.50 is by 274% greater than 19-Lot at Bid @2384.25: (71/19 - 1) * 100 = 274%.

Absorption is a diagonal imbalance but with level rejection. Absorption depth (in ticks) means how far price bounces from the level of absorption.
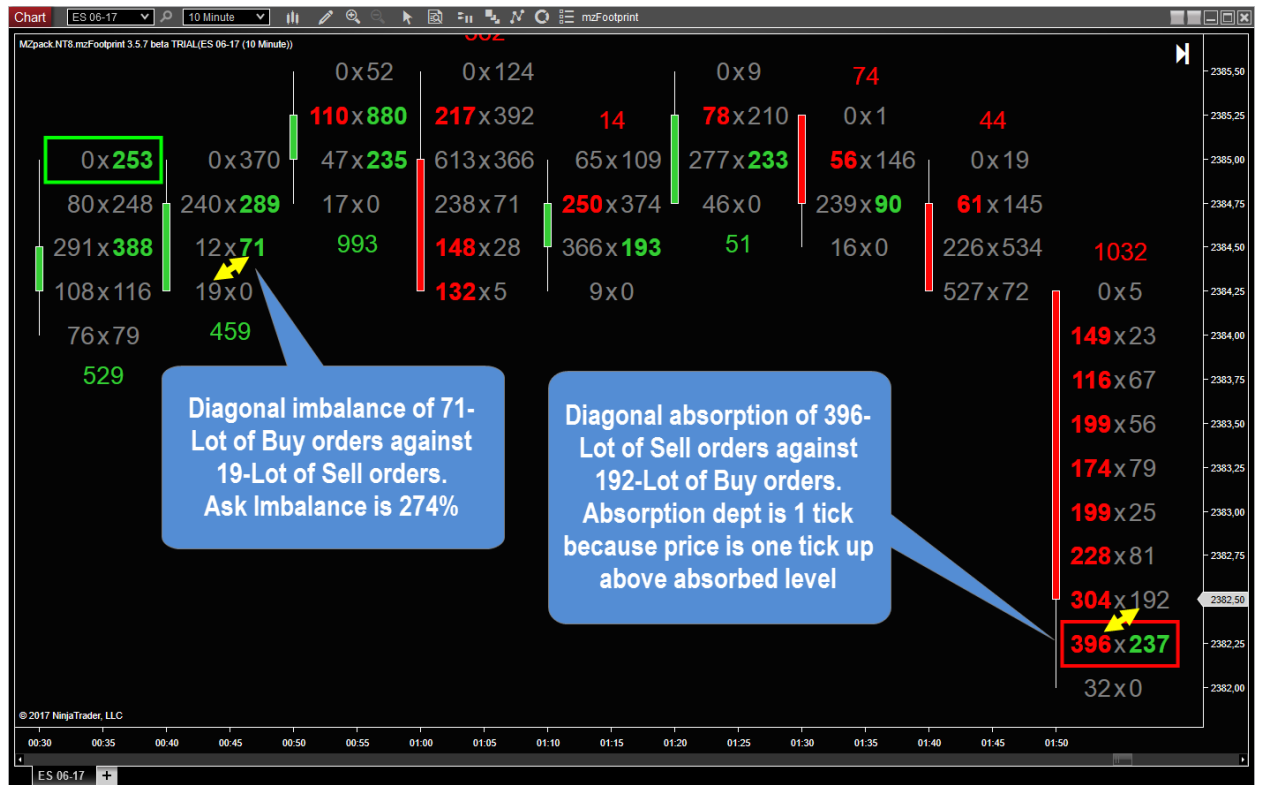
EXPLAINING COT

COT is the abbreviation for Commitment Of Traders. There are two COTs: COT High and COT Low.

What is COT? COT High explanation

COT High is cumulative bid/ask Delta starting from the moment when the price makes new high or repeats previous one. COT helps to see buy/sell balance after new price level has been reached.

We can consider new high as a market test whereas COT will be a reaction for it. If new (higher in this example) price is rejected by the market, we will see negative COT High price falling. If the price stays at highs and COT High is negative and is growing by modulo, then this means strong support by buy limit orders.

How To Use COT In Day Trading?

Read this article: https://www.mzpack.pro/trading-setups/what-is-cot-and-how-to-use-it-in-day-trading/

RATIO NUMBERS

Calculations. For up-bar Ratio is bid value above bar low divided by the bid at the bar low. For down-bar Ratio is ask value below bar high divided by the ask at the bar high.
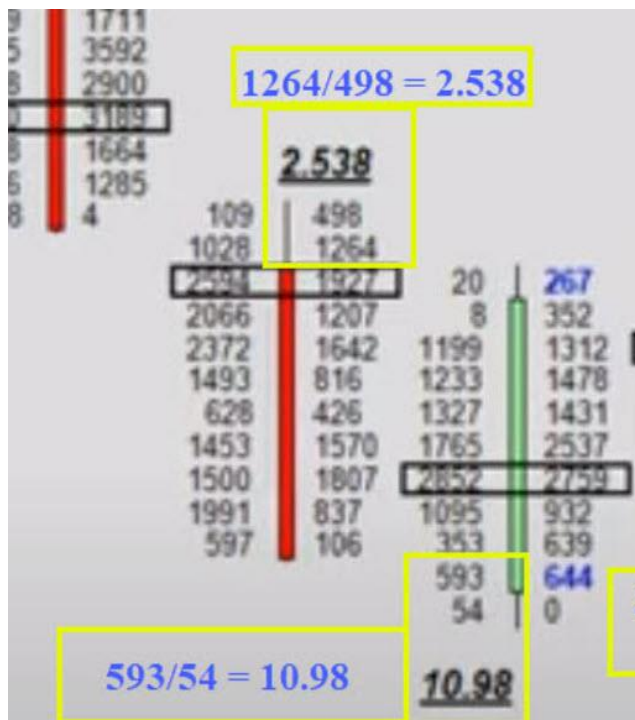
There are three types of ratios:

Neutral Ratio: which means the market is facilitating trade. A number between the high and low ratio (29.0 and 0.71 by default). If the number falls between these two numbers, the market is facilitating trade.

Ratio Bounds High: which indicates price is being REJECTED. A number above 29.0. So if the Ratio number prints below the bar (Up Bar) then lower prices are being rejected. And if the Ratio number prints above the bar (Down Bar) then higher prices are being rejected.

Ratio Bounds Low: which indicates a price level is being DEFENDED in the market. A number under 0.71 down to 0 (zero). So if the Ratio number prints below the (Up Bar) then the area is being supported by buyers. And if the Ratio number prints above the (Down Bar) then the area has sellers up there.

Read more on the forum:

https://mzpack.freshdesk.com/en/support/discussions/topics/48000558972/page/2?url_locale=en

SUPPORT AND RESISTANCE ZONES

S/R zones are based on levels of imbalances or absorptions. The more volume traded and more there are consecutive levels in a zone the stronger this particular zone is.

Levels of imbalances on ask side will create a support zone, whereas levels of imbalances on bid side will create a zone of resistance. In case of absorption, all are vice versa: levels of absorptions on ask side will create resistance zone, whereas levels of absorptions on bid side will create a zone of support.

Volume filter is a filter for total volume of support resistance zone including all levels inside zone.

Zone can be canceled on end of session (see 'Break on session' option) or if price crossed a zone and stays below resistance zone or above support zone.

Also, you can set alerts on zone appearing.

CLUSTER ZONES

Use **Cluster Zones** to spot specific values in the clusters. For example, Low Volume Nodes (**LVN**), High Volume Nodes (**HVN**), Delta ranges, Delta Percentage ranges, Trades ranges.

See settings for LVN
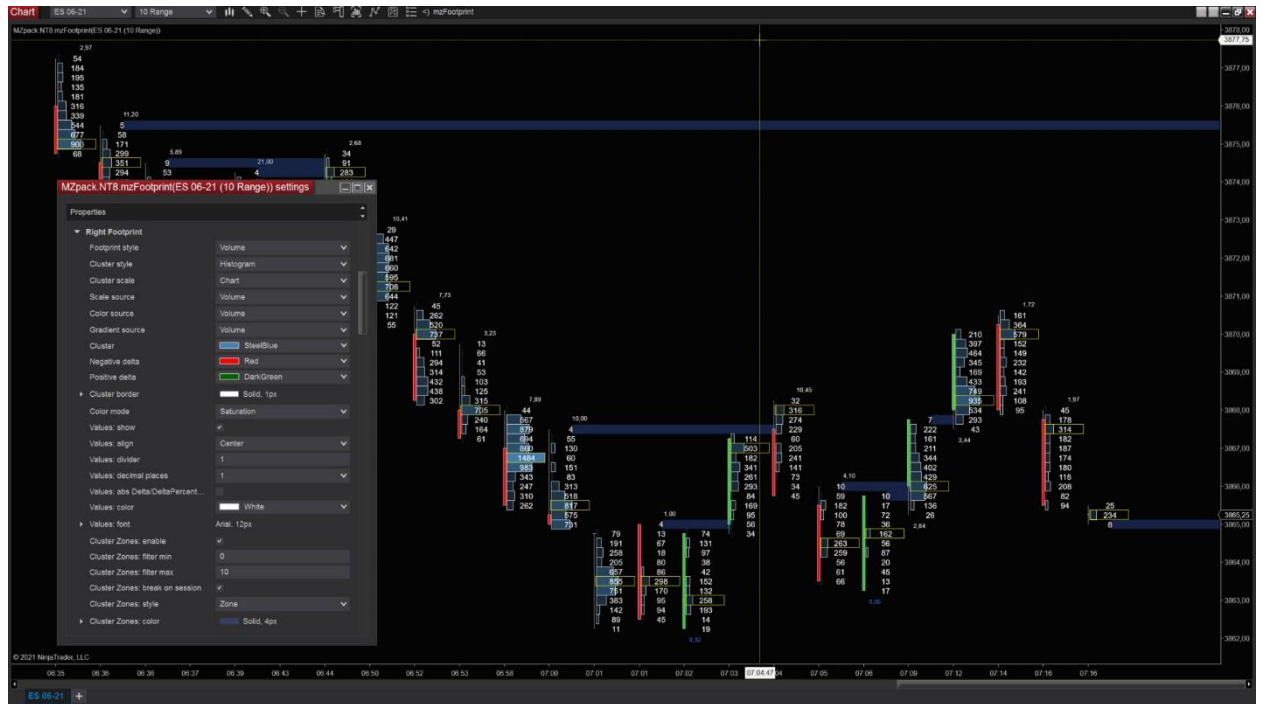
See settings for HVN to spot critical volumes
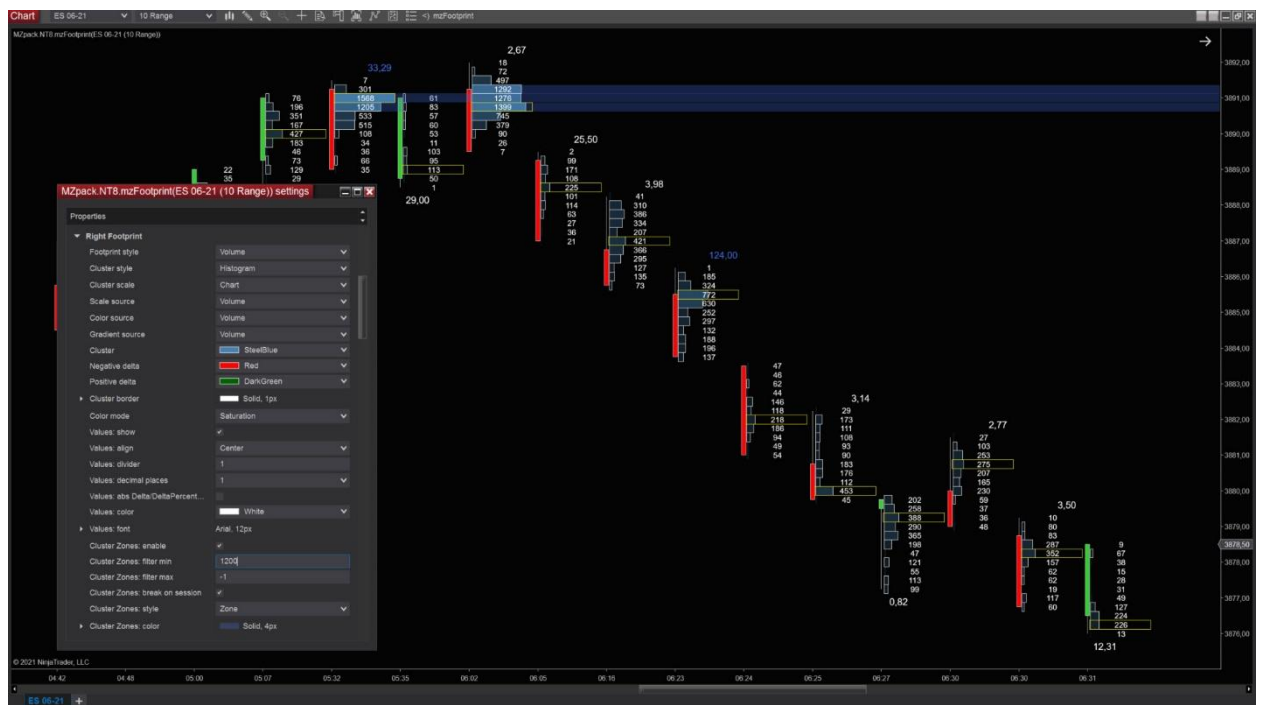
PREDICTED STAT VALUES

Stat values are being extrapolated (proportional to the bar time). The gauge shows the progress of the bar with time left to bar close. The feature is available for time-based intraday bar types only.

mzBigTrade Indicator

Video Guide (English) https://www.youtube.com/watch?v=OO_rWfxN-rQ

Video Review (German) https://www.youtube.com/watch?v=DVDhrqTg9rM

Indicator page: https://www.mzpack.pro/product/mzbigtrade-for-ninjatrader-8/

Indicator settings

Filters

- **Logic**. Filtering logic**.** ANY – any of enabled filters must be met. ALL – all of enabled filters must be met
- **Trade: enable.** Enable filtering by trade volume
- **Trade: min volume**. Minimal trade volume
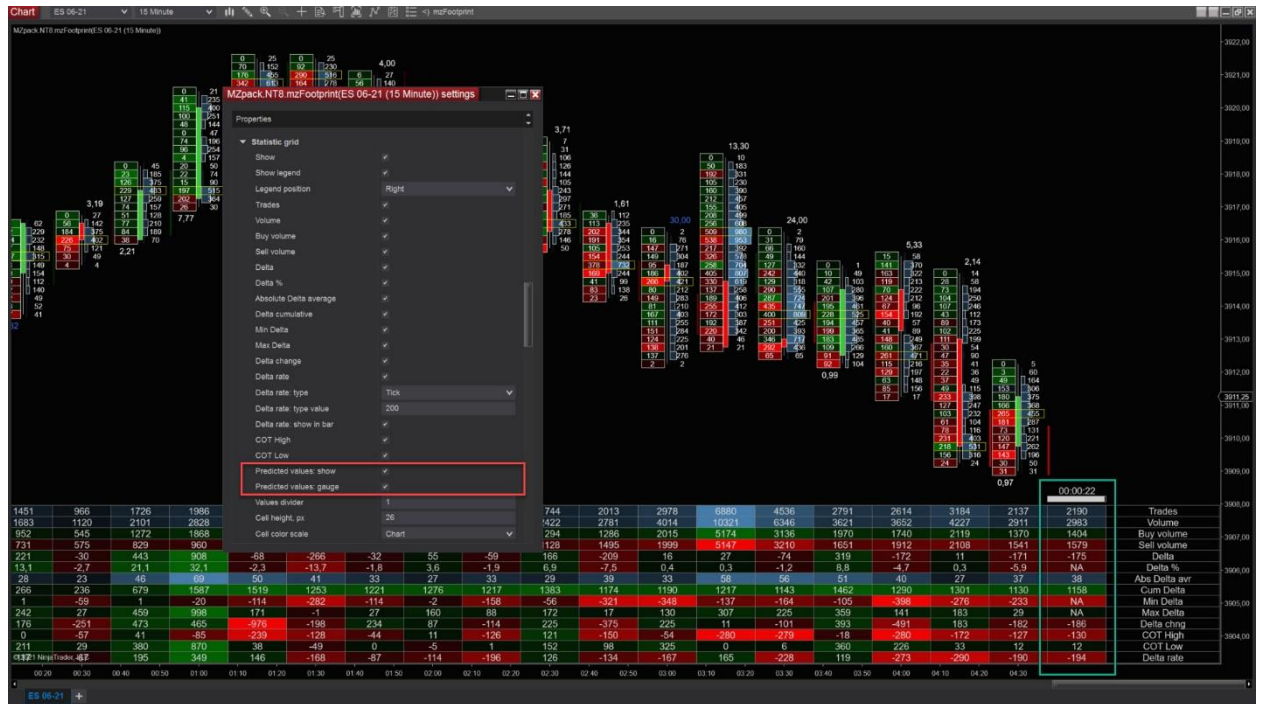- **Trade: max volume**. Maximal trade volume. Set -1 for unlimited volume.
- **Iceberg: enable.** Enable filtering by iceberg volume
- **Iceberg: algorithm**. *Iceberg detection works on live or Market replay data only*
    a.   Hard
    b.   Soft
- **Iceberg: min volume**. Minimal iceberg volume
- **DOM pressure: enable.** Enable filtering by DOM pressure. DOM pressure might be a negative value. This filter uses the absolute value of DOM pressure for comparing. E.g. -100 will pass the filter = 90 because abs(-100) > 90
- **DOM pressure: min volume**. Minimal DOM pressure volume.
- **DOM pressure: max volume**. Maximal DOM pressure volume. Set -1 for unlimited volume
- **DOM support: enable.** Enable filtering by DOM support
- **DOM support: min volume**. Minimal DOM support volume.
- **DOM support: max volume**. Maximal DOM support volume. Set -1 for unlimited volume

Extra Filters

- **Aggression: enable.** Enable filtering by aggression. Aggression means range of the trade (or sweep range) measured in ticks
- **Aggression: min, ticks**. Minimal aggression range
- **Aggression: max, ticks**. Maximal aggression range. Set -1 for unlimited range.

- **Smart/Predatory: enable**. Enable filtering by 'smart' trades.

Presentation

- **Max number of trades in chart frame**. See *Filters and Auto-filtering* topic.
- **Marker**. Big trade marker
    - **a.** **Line**
    - **b.** **Bubble**
    - **c.** **Box**
    - **d.** **Bar**
- **Position**. Position of big trade marker
    - a. **First**. First tick of the trade
    - b. **Last**. Last tick of the trade
    - c. **Hi**. Highest price of the trade
    - d. **Lo**. Lowest price of the trade
    - e. **POC**. Point Of Control
    - f. **Stacked**. See *Stacked Tape* topic
- **Buy line**. Line for Buy trades in Line marker mode
- **Sell line**. Line for Sell trades in Line marker mode
- **Min line length**. Lines lengths are proportional to trades volumes and will be between Min and Max length
- **Max line length**
- **Scale line thickness**. Set maximal line thickness by changing Buy/Sell line Width value. All lines thicknesses will be scaled proportionally to trades volumes
- **Min shape extent**. Bubble and Box extents are proportional to trades volumes and will be between Min and Max extents
- **Max shape extent**
- **Buy shape color**
- **Sell shape color**
- **Iceberg color**. Color of shape border for trades with iceberg orders. *Iceberg order is marked with contour if iceberg value is greater or equal to the iceberg filter even Iceberg filter is disabled*
- **Color mode**
    - a. Solid. Selected solid color for all data
    - b. Saturation. Saturation depending on volume/delta values
    - c. Heatmap. Heatmap colors by volume/delta values
    - d. GrayScaleHeatmap. Gay scaled heatmap colors by volume/delta values
- **Saturation preset.** 4 levels of saturation for trades markers

- **Show DOM pressure**. Display DOM pressure with triangles. *Triangle is displayed if corresponded volume is greater or equal to the filter even DOM pressure filter is disabled*
- **Show DOM support.** Display DOM support with triangles. *Triangle is displayed if corresponded volume is greater or equal to the filter even DOM support filter is disabled*
- **Liquidity pushing/DOM support triangle thickness.** 5 predefined levels of thickness
- **Liquidity pulling line.** Line for contour of tringle displaying liquidity pulling (negative DOM pressure)
- **Show trade POC**. Show Point of Control of the trade by horizontal line.
- **Volume position**. Volume value alignment relative to shape: Inside, OutsideLeft, OutsideRigth, None
- **Volume text font**
- **Buy volume in shape color**. Color for volume value inside Bubble/Box
- **Sell volume in shape color**. Color for volume value inside Bubble/Box

Other

- **Draw lines on right margin**. Only for Line marker mode
- **Show popup info**
- **Order ticks**. Order ticks by volume in popup information. If not checked ticks will be ordered by time as they come in order flow

Notifications

- **Alert**. Alert for big trade passed your filters
- **Buy sound**
- **Sell sound**
- **Send Email**
- **Email address**

FILTERS AND AUTO-FILTERING

Manually set filters and auto-filtering allow to control the presentation of order flow on the chart.

Pay attention to applying of manual filters. We have two type of logic for combining filters.

- ANY – any of enabled filters must be met
- ALL – all of enabled filters must be met

Also, indicator has Extra Filters: Aggression and Smart/Predatory. These filters applied over the main filters.

Auto-filtering feature allows easily spot significant trades on the chart. It works in the background. Use **Presentation - Max number of trades in chart frame** parameter to adjust theintensity of the order flow on the chart.
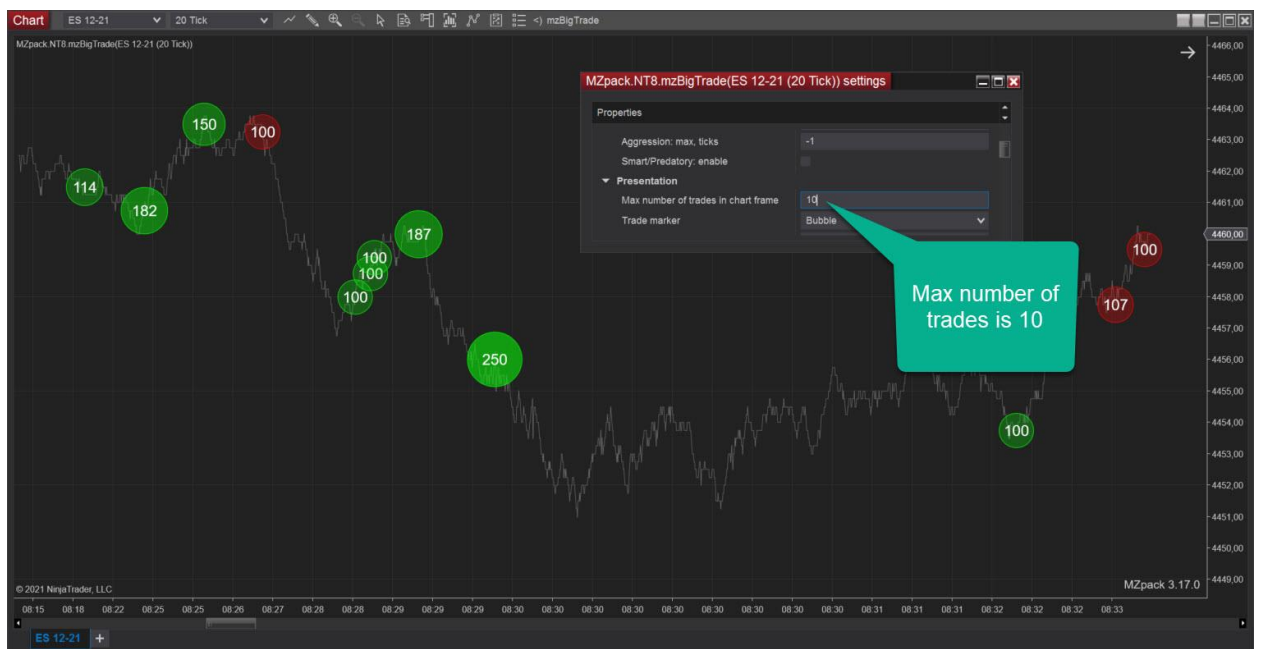
When the zoom is increased for the chart, smaller and smaller trades become visible because those small trades make sense on zoomed (smaller) chart frame. Thanks to auto-filtering you can dig into order flow as soon as you need it.

Size of iceberg part of the order, dom pressure, and dom support are also considered in auto-filtering algorithm during the selection of trades.

Note, that the scale and color intensity of trades markers is calculated on all loaded market data.

STACKED TAPE

To enable Stacked tape choose Stacked for **Presentation - Marker position** parameter.

The trades are organized vertically, by bar like a chain, and in the sequence in that trades come from the order flow. New buy trades are being added at the top, and new sell trades - at the bottom of each chain.

Stacked tape is extremely useful when you work with non-reconstructed tape, or when you have a lot of small trades for the traded instrument.

We recommend using stacked position on tick time frames like 20 Ticks.

Note, that stacked position is available for Bubble and Box markers only.
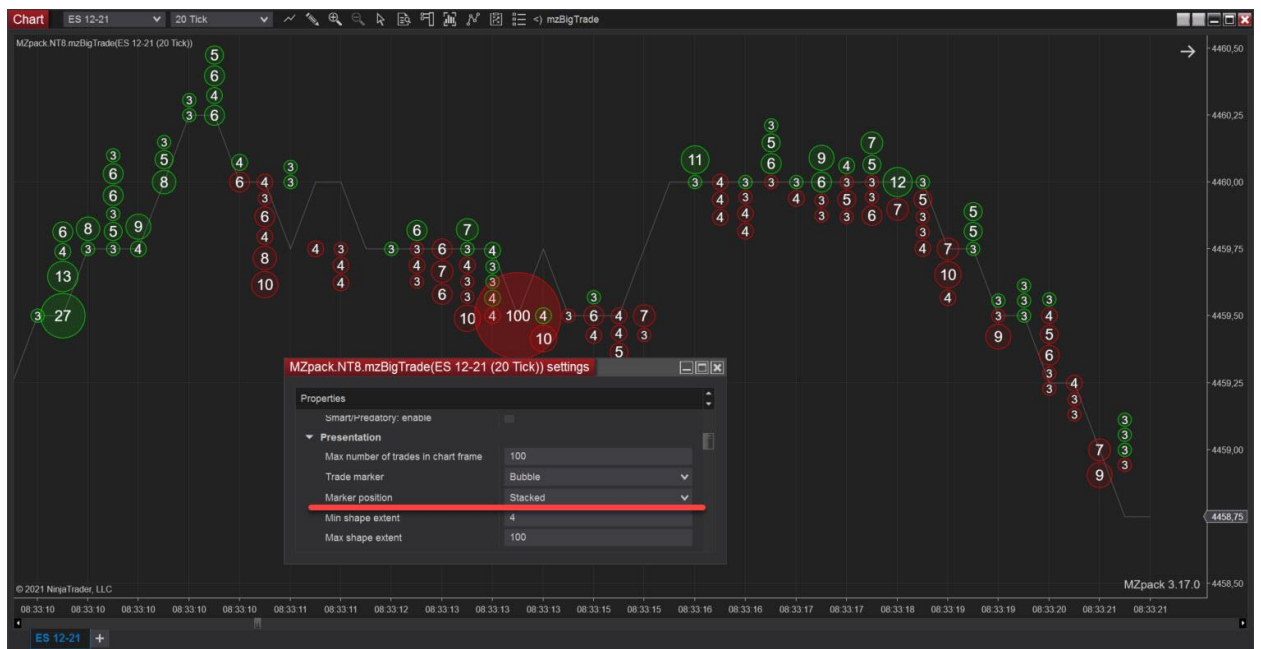
POP-UP INFO

The pop-up info shows all parameters of a trade in small floating window. The pop-up info is enabled by default. To disable this feature uncheck **Other – Show pop-up info**.
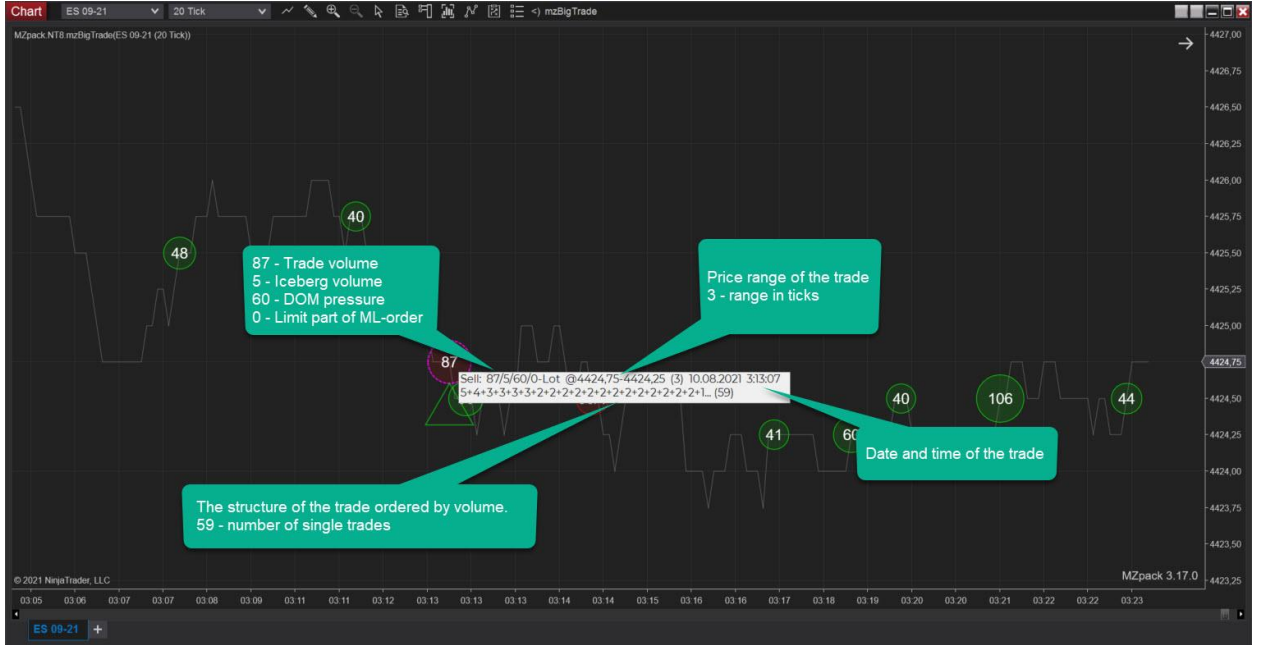

More examples
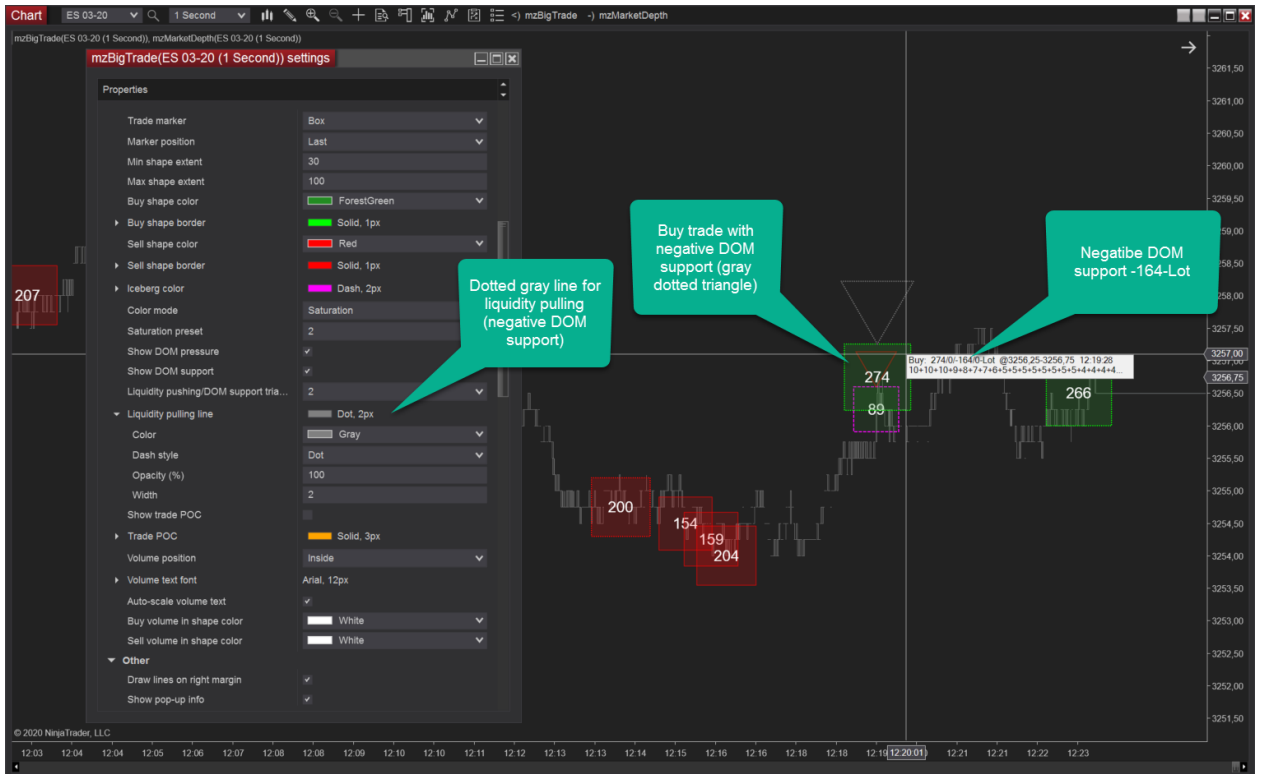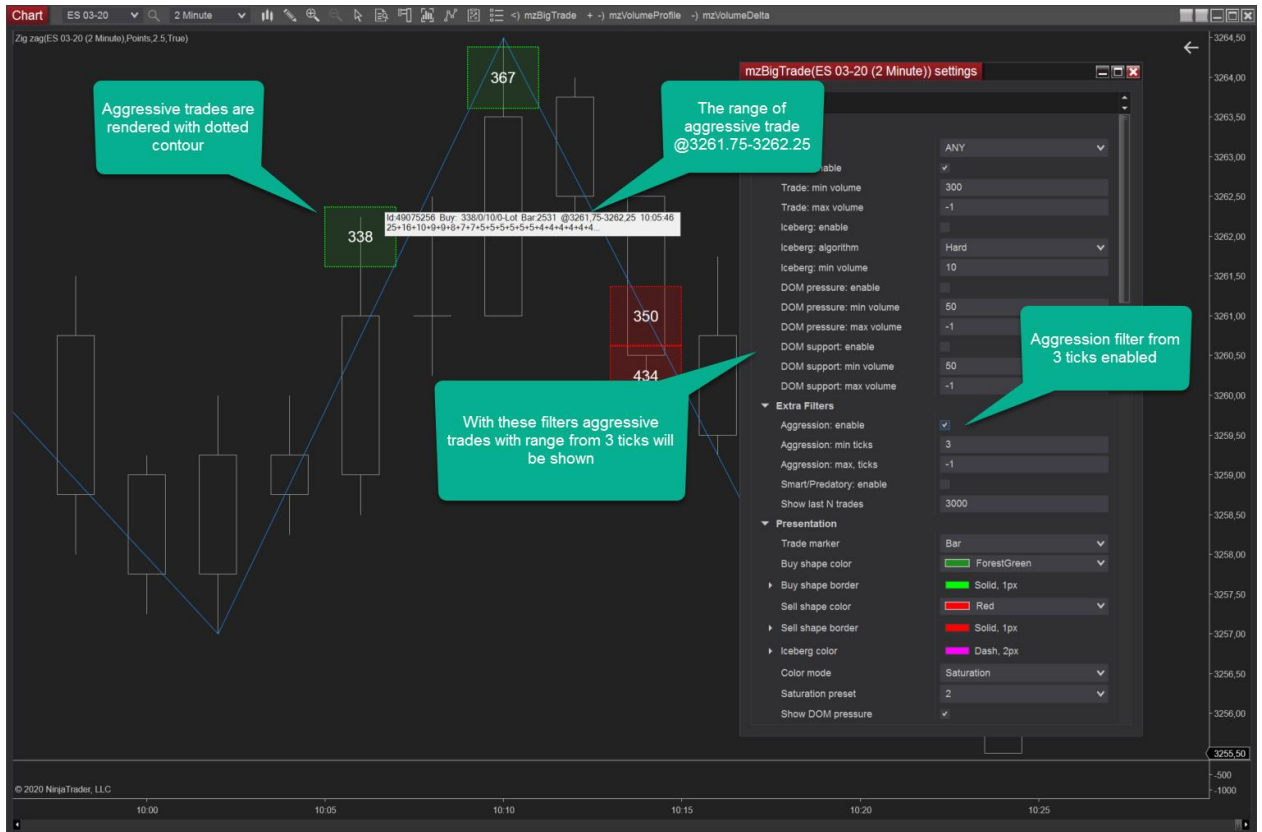
ICEBERG ALGORITHMS

Iceberg search algorithms work only on live or Market replay data only. An iceberg order is a special limit order. The main size of an iceberg order is not shown in the order book. You will never know size of the hidden part (iceberg) of limit order without analyzing algorithm running in real-time.

Hard

In this mode we use volumes on best bid/ask price levels to calculate the size of the hidden liquidity

Soft

In this mode we add positive DOM pressure (if any) to the size of the iceberg order calculated by Hard algorithm.
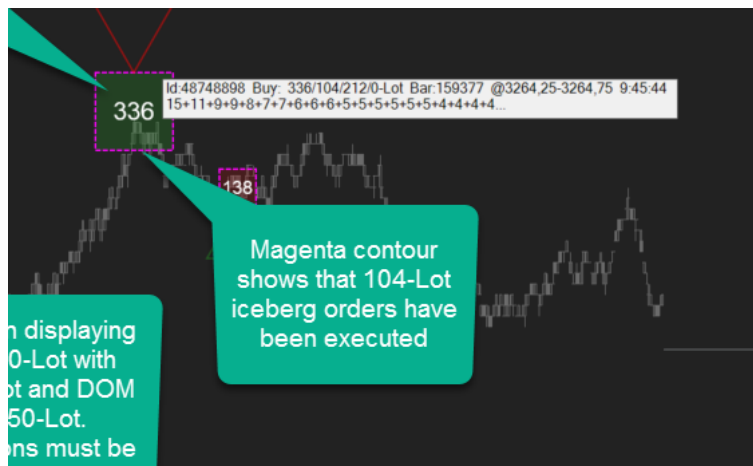
DOM PRESSURE

We use *DOM pressure* term to describe behavior of the liquidity on the way of market order.

Any trade you see in mzBigTrade indicator, is the result of matching market order with limit order (liquidity) resting in order book or DOM. This matching occurs for the best Bid or Ask prices depending on direction of the order – sell or buy.

DOM pressure can be negative or positive.

Positive DOM pressure means **adding liquidity** on the best Bid or Ask. **Negative DOM pressure** means **removing liquidity** from the best Bid or Ask price.

Eg. buy trade with positive DOM pressure (see image: green box with red triangle) means adding of the liquidity on best Ask upon execution of the order. For sell order with DOM pressure liquidity is being added at the best Bid price. Positive DOM pressure always occurs at the opposite side of the order and is a resistance to the trend.
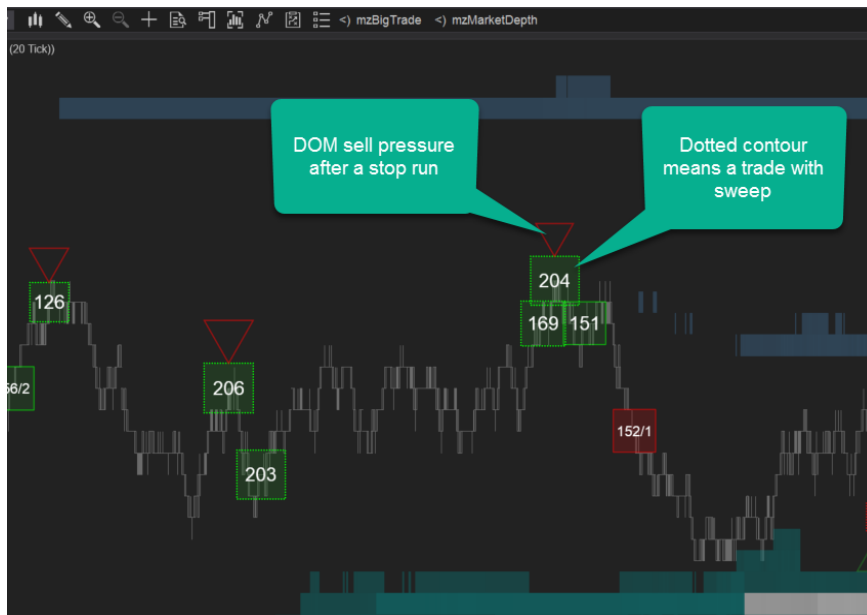
Here is an example of negative DOM pressure occurred on the offer side (best Ask) of the DOM after a buy trade. Negative DOM pressure always occurs at the opposite side of the order and means that resistance to the trend is becoming weaker.
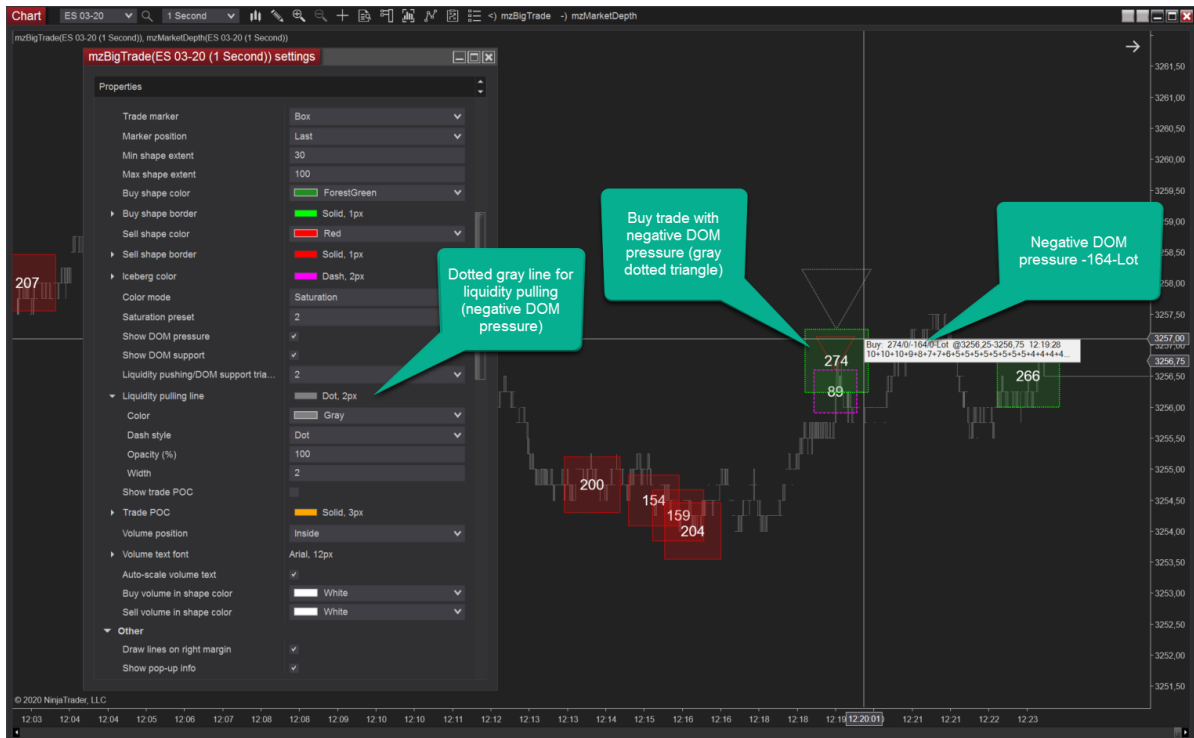
DOM SUPPORT, MARKET-LIMIT ORDERS

We use *DOM support* term to describe behavior of the liquidity in the tail of executed market order. See image. In this example buy trade has DOM support 50-Lot at the bid side. These 50 contracts have been added at the best Bid price right after execution of the order. Note, that both the trade marker (green box) and the triangle have the same color.
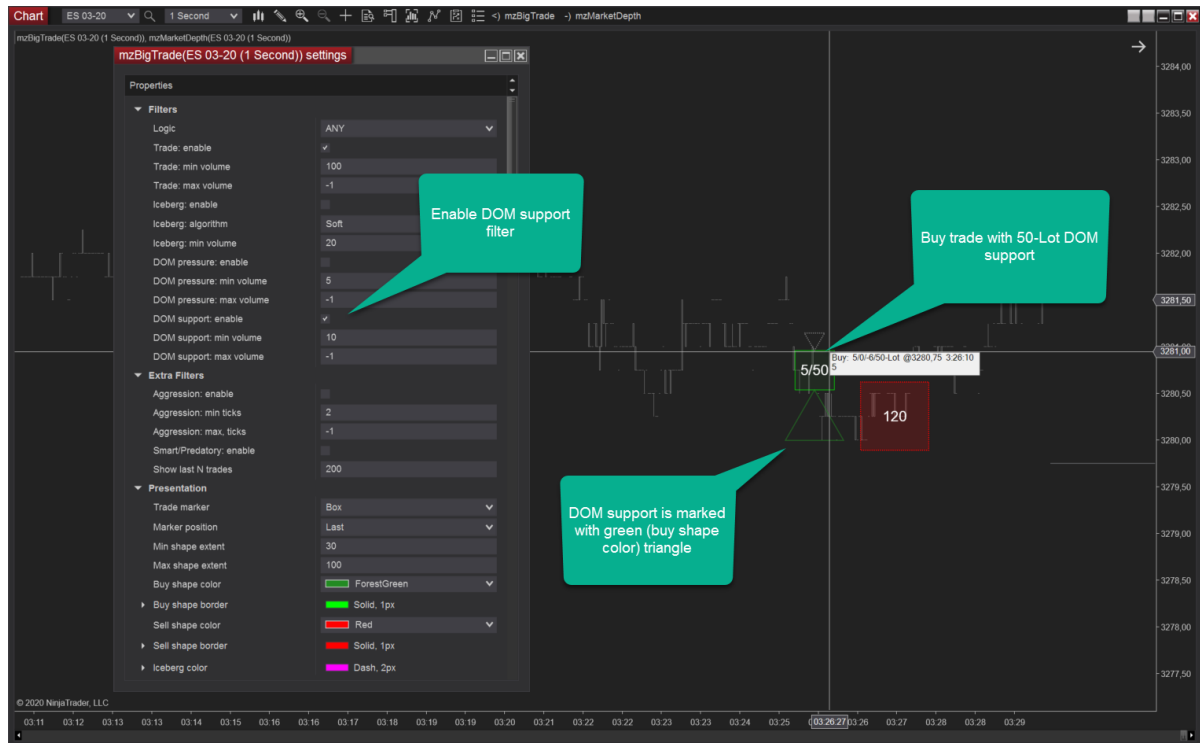
Big or 'smart' money and institutional traders often act by Market-Limit and Stop-Limit orders. Market-Limit orders are executed at the best price available on the market. If the Market-Limit order can only be partially filled, the order becomes a limit order and the remaining quantity remains on the order book at the specified limit price. Learn more at CME Group Wiki

To spot Market-Limit orders enable 'DOM support' filter. Also, enable Extra Filters: 'Smart' and 'Aggression' = 1

In case of buy Market-Limit order limit part of the order will be a support for the price whereas for sell Market-Limit order limit part will be a resistance.

AGGRESSIVE TRADES

An aggressive trade eats liquidity on two or more price levels to get filled. An aggressive trade can be initiative market order or triggered stop-market orders (stop run). Also, aggressive trade can be called a trade with sweep.

A stop-market order lets you send a market buy or sell order once the market has reached a specific price (the stop price). When the market price reaches the stop price, the order gets triggered automatically and a market order is sent.

Aggressive trades are shown with dotted contour. See image.
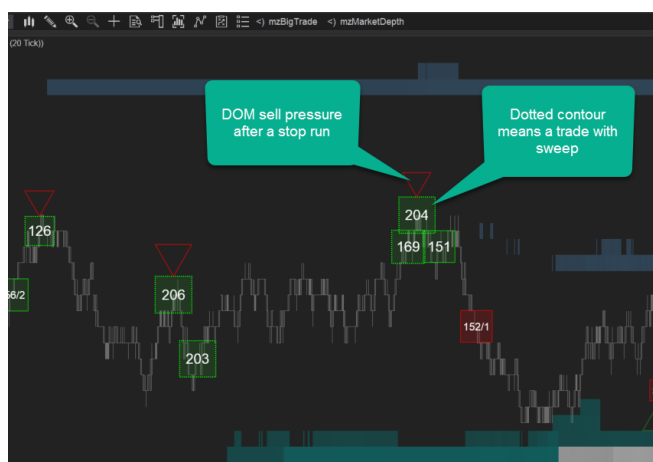
Use Bar for 'Trade marker' parameter to cleary see range of the trade
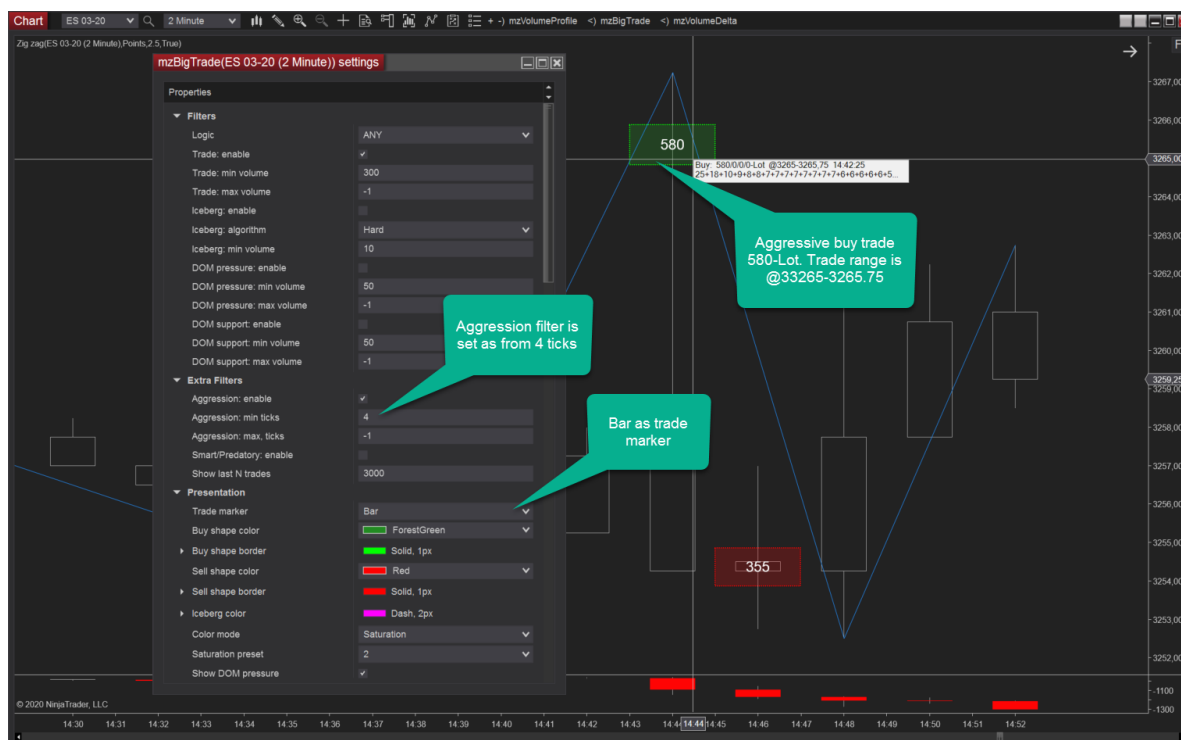
SMART/PREDATORY TRADES

Smart/Predatory trade eats all available liquidity on best bid or ask. The amount of smart/predatory trades depends on the instrument.
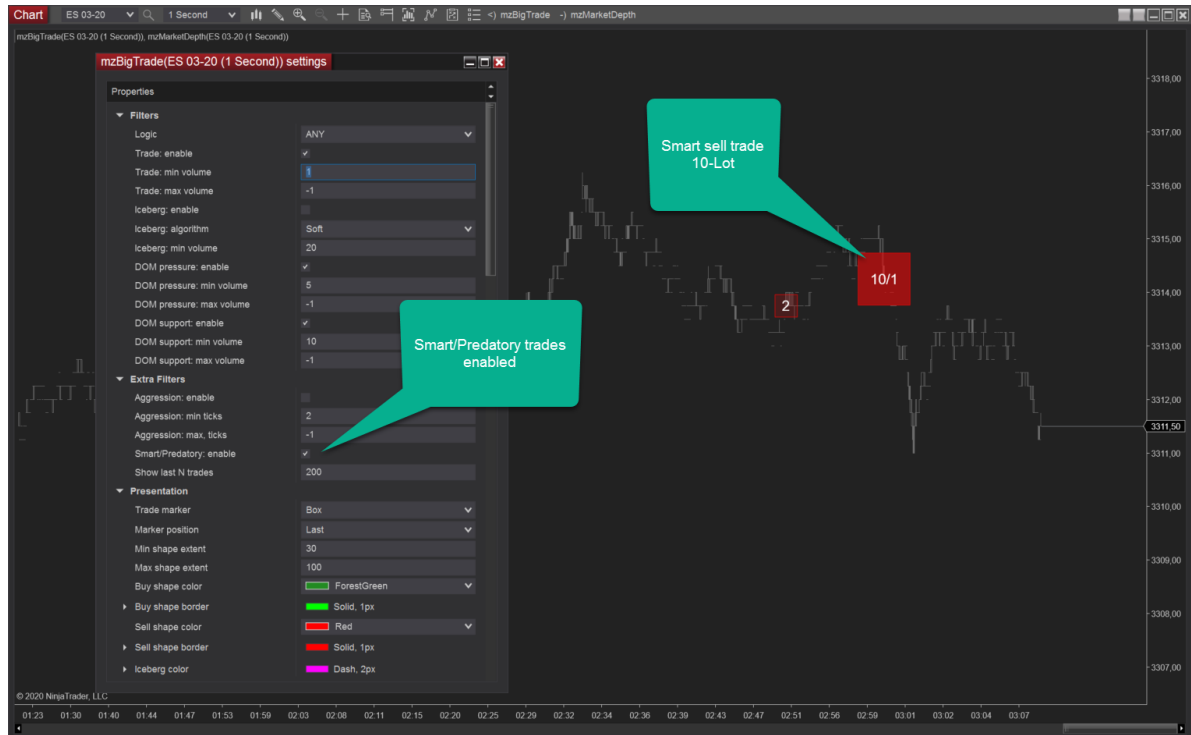
RECONSTRUCT TAPE MODE

MZpack Order flow core is able to reconstruct single (tick) trades into aggregated trades. Enable 'Reconstruct tape' option (default setting) to see aggregated trades. You can change this option On-The-Fly. See **Orderflow Settings** topic.

Note! Using OnTheFly method or disabling 'Reconstruct tape' will significantly increase memory consumption.

mzVolumeProfile Indicator

Video Guide (English) https://www.youtube.com/watch?v=RBBrWeexU28

Video Review (German) https://www.youtube.com/watch?v=DVDhrqTg9rM

Indicator page: https://www.mzpack.pro/product/mzvolumeprofile-for-ninjatrader-8/
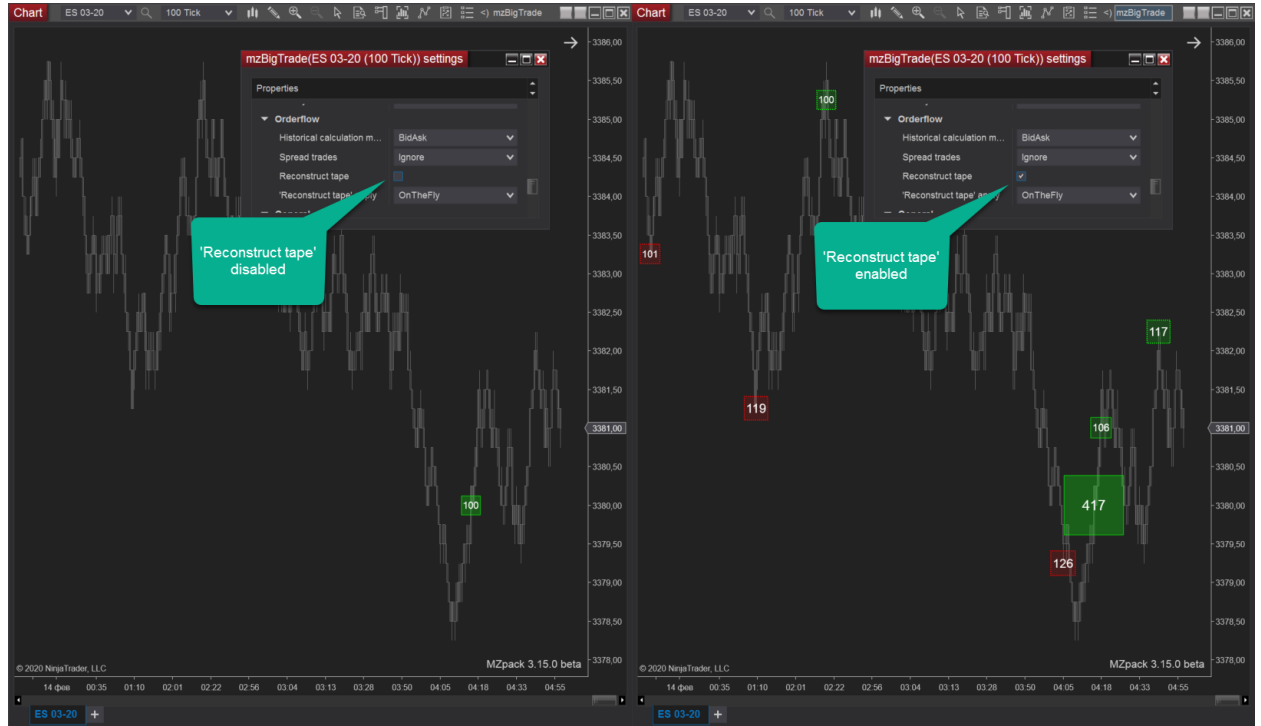
Indicator settings

Common

- **Show Chart Profiles**
  - a. **TPO**. Show TPO or Market Profile
  - b. **VP.** Show Volume profile
  - c. **ALL.** VP + TPO
  - d. **None**
- **Chart Profile mode**
  - a. **Bar.** Build profile per each bar
  - b. **Session**. Build profile for each session
  - c. **Custom**. Custom ranged volume profile
  - d. **Composite**. Cumulative volume profile for all chart data
  - e. **Bars**. Build profile each given value of bars. See 'Profile creation value'
  - f. **Daily.** Build profile for each day
  - g. **Weekly.** Build profile for each week
  - h. **Monthly.** Build profile for each month
  - i. **Quarterly.** Build profile for each quarter
  - j. **Yearly.** Build profile for each year
  - k. **Sessions.** Build profile for each given value of sessions. See 'Profile creation value'. Applied for Stacked profiles only.
  - l. **Bars.** Build profile for each given value of bars. See 'Profile creation value'.
  - m. **Days.** Build profile for each given value of days. See 'Profile creation value'. Applied for Stacked profiles only.
  - n. **Weeks.** Build profile for each given value of weeks. See 'Profile creation value'. Applied for Stacked profiles only.
  - o. **Months.** Build profile for each given value of months. See 'Profile creation value'. Applied for Stacked profiles only.

    p.   **Volume**. Build new profile each time **profile volume** reaches given value. See 'Profile creation value'

    q.   **Delta**. Build new profile each time **absolute delta** reaches given value. See 'Profile creation value'

    r.   **Tick.** Build new profile on each given number of ticks or trades. See 'Profile creation value'

    s.   **Continuous.** Use this mode with **Working time** enabled.

    t.   **None.**

- **Profile creation value**. Value to control creation of custom periodical profiles: *Bars, Sessions, Days, Weeks, Months, Volume, Delta, Tick*.
- **Session break**. Break profile by session time for Bars profile mode
- **Profile accuracy.** Tick or Minute. See details below.
- **Ticks per level**. Each level of profile contains this number of price ticks
- **Border**. Show profile border
- **Border**. Border line
- **Background**. Show profile background
- **Background**. Color for profile background
- **Background opacity, %**.
- **Buttons font**

TPO

- **Letter period, min.**
- **Presentation.** Letter ot Block. When there is no room to render TPO letters, blocks are rendered
- **Split.** Split TPO periods
- **IB, min.** Initial balance in minutes
- **Show IB**
- **IB color**
- **Max width, %.** Maximal width of TPO in % of total profile range.
- **Position.** TPO position: Left, Rigth or RightOnChartMargin
- **Rotate colors.** If unchecked Letter/block color 1 is used for all TPO letters
- **Letter/block color 1**
- **Letter/block color 2**
- …
- **Letter/block color 6**
- **Letters font**
- **Show period open**. Render O at the beginning of each TPO period
- **Open period color**
- **Show POC**

- **POC color**
- **Show VA.** Show TPO value area
- **VA, %**
- **VA color**
- **POC/VA opacity, %**

TPO levels

- **#1 Enable.** Enable group #1 of levels
- **#1 Attach to.** Attach group of levels to All profile or to profile with given T. For Custom profile mode 'All' value is only available.
- **#1 Attach to T - .** The absolute index of the profile to which the levels are attached. The last profile has T = 0, the profile before the last profile has T = 1
- **#1 Naked by.** A level is naked by Any profile, by the Next profile or by profile with given T index. Naked by T is available in Custom profile mode also.
- **#1 Naked by T - .** The absolute index of the profile which may break a level. In Custom profile mode the last added custom profile has T = 0.
- **#1 Value position**
- **#1 Open.** Level at open price of profile: Disabled, Extended, Naked
- **#1 Close.** Level at close price of profile: Disabled, Extended, Naked
- **#1 High …**
- **#1 Low**
- **#1 IB High**
- **#1 IB Low**
- **#1 TPO POC**
- **#1 TPO VAL**
- **#1 TPO VAH**
- **#2 Enable …**
- **#3 Enable …**

Volume Profile

- **Profyle type**
  - a.   **Volume**. Total volume histogram
  - b.   **BuySell**. Buy and sell volumes histogram
  - c.   **Delta**. Bid/Ask delta histogram
  - d.   **VolumeDelta**. Similar to Volume type but ladders are colored with delta colors.
  - **e.   None**
- **Profile view**. Ladders, Area, Contour.

- **Show last N profiles**. Set 1 to display the last profile only, etc. Set -1 to display all profiles.
- **VA, %**
- **Position**. Volume profile position – Left, Rigth or RightOnChartMargin
- **Width, %.** Width of profile histogram. Negative value can be used, eg. -50%, and the profile will be mirrored.
- **Relative width**. Scale the width of profiles according to chosen type of values among all chart profiles
- **VA color**. Value Area color
- **VAH/VAL color.** Profile histogram color
- **Buy color**. Buy volume histogram color
- **Sell color**. Sell volume histogram color
- **Opacity, %.** Opacity of ladders, area, etc.
- **Ladders: saturation**
- **Values: show.** Show choosen values for profile ladders
  - a. Volume. Total ladder volume
  - b. Delta. Bid/Ask delta
  - c. BidAsk. Bid x Ask notation
  - d. None
- **Values: 1K divider.** Divide volume and delta values by 1000.
- **Values: color**. Color for values.
- **Values: font**. Font for ladders values. Font size will be automatically scaled to fit ladder height.
- **Delta: show**. Display delta histogram over profile ladders
- **Delta: % of profile width**
- **Delta: positive**. Color for positive delta
- **Delta: negative**. Color for negative delta
- **Delta: opacity, %**
- **Delta: saturation**
- **Profile: developing border**. Border line for developing profile.
- **Profile: ladders border**. Ladders border style and color
- **Profile: labels font**.
- **Profile: right margin, px**
- **Custom profile: unique settings**. If checked, each custom profile has unique settings accessible via UI buttons. UI buttons are visible when profile is selected with the mouse.
- **Custom profile: add = Mouse Left +.** A keyboard key. Use this combo for adding a profile

Volume Profile levels

- **POC: mode**
  - a. Off
  - b. On
  - c. Naked
  - d. Extended
  - e. Developing
  - f. DevelopingNaked
- **POC: line.**
- **VAH/VAL: mode**
  - a. Off
  - b. On
  - c. Naked
  - d. Extended
  - e. Developing
  - f. DevelopingNaked
- **VAH/VAL: line**.
- **Levels values**. Show choosen values for profile levels: Price, Volume, Delta, None

Volume Profile VWAP

- **VWAP: mode**
  - a. Last. Last value of VWAP as horizontal line
  - b. Dynamic
  - c. DynamicStdDev1. Dynamic VWAP with standard deviation #1
  - d. DynamicStdDev2. Dynamic VWAP with standard deviation #2
  - e. None
- **VWAP: line**
- **VWAP: Sigma 1.** Sigma value for standard deviation #1. Use 1 for 1st standard deviation
- **VWAP: Std deviation #1.** Line for Sigma 1 deviation
- **VWAP: Sigma 2.** Sigma value for standard deviation #2. Use 2 for 2nd standard deviation
- **VWAP: Std deviation #2.** Line for Sigma 2 deviation

Volume Profile statistics

- **Total volume**. Total volume of the profie
- **Delta**. Display mode of profile's delta
  - a. Delta
  - b. BidAsk
  - c. None

- **High/Low**
- **Range, ticks**
- **POC**
- **VAH/VAL**
- **Volume Profile VWAP**
- **Begin/End**. Begin and end time of the profile

Stacked profiles. See STACKED PROFILES topic

- **#1: show.** Show stacked profile #1: TPO, VP, ALL or None
- **#1: mode**. See '**Common – Chart Profile mode**'.
- **#1: creation value**. Value to control creation of custom periodical profiles. See '**Common – Profile creation value'**.
- **#1: type**. See '**Volume Profile – Profile type**'.
- **#1: presentation**. Ladders, Area or Contour
- **#1: profile POC/VAH/VAL mode**. On, Off, Naked, Extended, Developing, DevelopingNaked
- **#1: POC**. Line for Point of Control
- **#1: VAH/VAL**. Line for VAH/VAL
- **#1: Delta:** Show Delta ladders over profile
- **#2: show.**
- …
- **Delta width, %.** Maximal width of Delta ladders. Enable, for example **#1: Delta**
- **Left margin, px**. Left margin of Stacked profiles group in pixels.
- **Rigth margin, px**. Right margin of Stacked profiles group in pixels.
- **Legend**. Show profile legend containing profile mode and N last/ago value.

A custom profile will be a developing profile if it contains the last bar of the chart.

CALCULATE MODES AND PROFILE RESOLUTION

mzVolumeProfile supports OnBarClose mode for optimization of system resources. There are three sets of settings.

Operation	Calculate	Tick replay	Profile accuracy	VP resolution	TPO resolution
Normal	OnEachTick	Enabled	Tick	1 Tick	1 Minute
Fast	OnBarClose	Enabled	Tick	1 Tick	1 Minute
The Fastest	OnBarClose	Disabled	Minute	1 Minute	1 Minute

Use fast and the fastest settings for such intensive order flow like BTCUSD, etc., and for large volume profiles. Note, that other combinations of settings are not supported by the indicator!

PROFILE ACCURACY

https://www.youtube.com/watch?v=8PdO2igdK10

Profile update and calculation accuracy affects on how long it takes to calculate profiles on loaded data. Use **Profile accuracy = Minute** with **disabled Tick replay** to speed up calculation of Volume Profiles and TPO for huge periods like months or years. In this mode, you don't need expensive historical tick data.

Use **Tick accuracy** and **Tick replay** mode enabled if you need precise Delta information with **Days to load = 1..14** (depending on the performance of your machine).

Tick

Set **Profile accuracy** to 'Tick' to get maximal precision of Delta on historical data. Enable Tick replay option from Data Series properties.

If Tick Replay is disabled, you will see a warning message and historical plots are not shown.

Minute

Set **Profile accuracy** to 'Minute' to speed up chart loading time up to x100 times. **Bars period type for the chart must be 'Minute'**. Disable Tick replay option from Data Series properties.

If Tick Replay is not disabled you will see a warning message and data (historical and real-time) are not processed.

HISTORICAL DATA

Minute volumes are being spread proportionally across each minute range. Volumes of up bars are buy volumes, volumes of down bars are sell volumes. For strategies API: the calculation of the profiles will be OnBarClose - this means 'on close of each 1 Min interval according to Profile Accuracy.

REAL-TIME DATA

Orderflow Calculation mode is 'UpDownTick'. The rest is the same as for Tick accuracy

Orderflow settings are not available for Minute accuracy.
We recommend **setting 'Ticks per level' = 10..20**.

NOTE

To emulate **Daily bars** use **1440 Minute bars**.

TPO LEVELS

TPO levels consist of three groups of levels. Each group can be attached to all chart profiles or to a profile with given T index. The last (current) profile of the chart has T index = 0, the previous profile has T index = 1, etc.

TPO levels include a set of significant profile levels: Open, Close, High, Low, IB High, IB Low, TPO POC, TPO VAL, TPO VAH. Each level can be Extended or Naked. Naked level can be terminated by any profile, next profile or a profile with given T index.
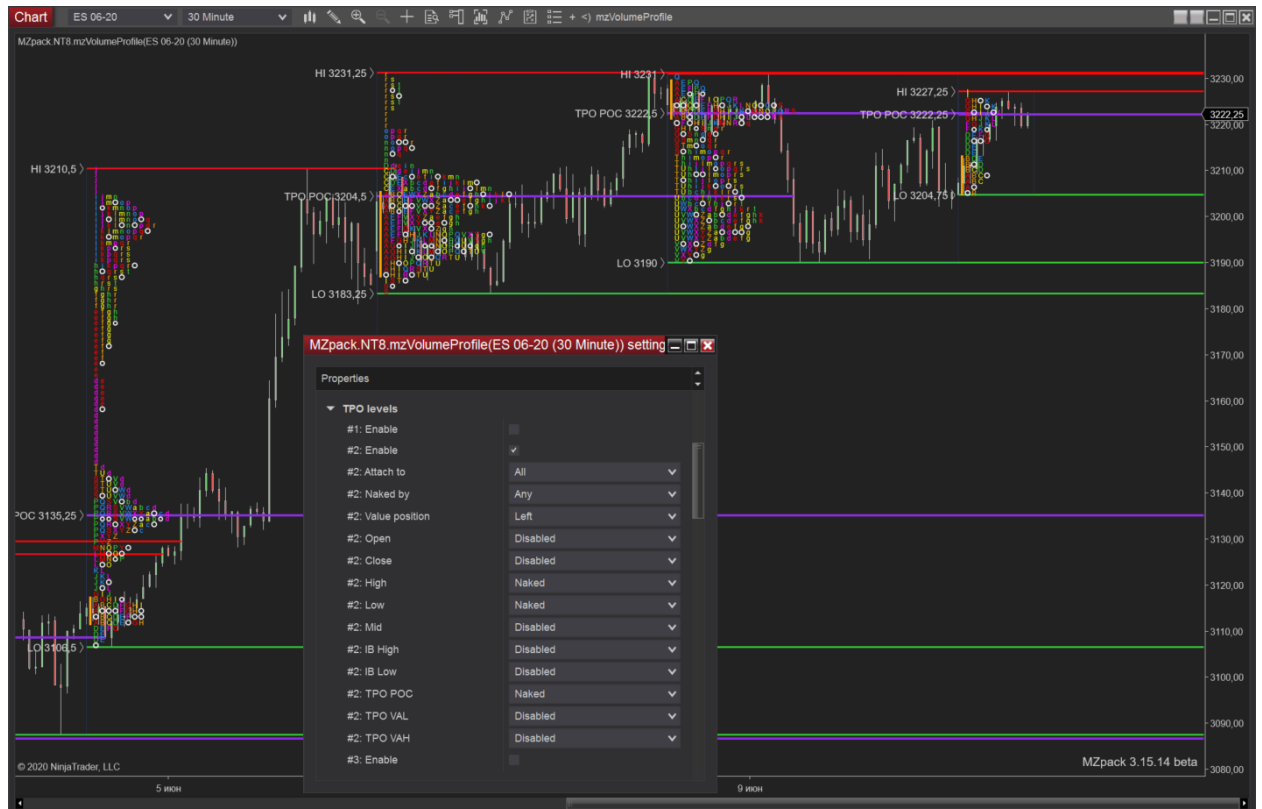
Some TPO levels can be used with volume profiles.
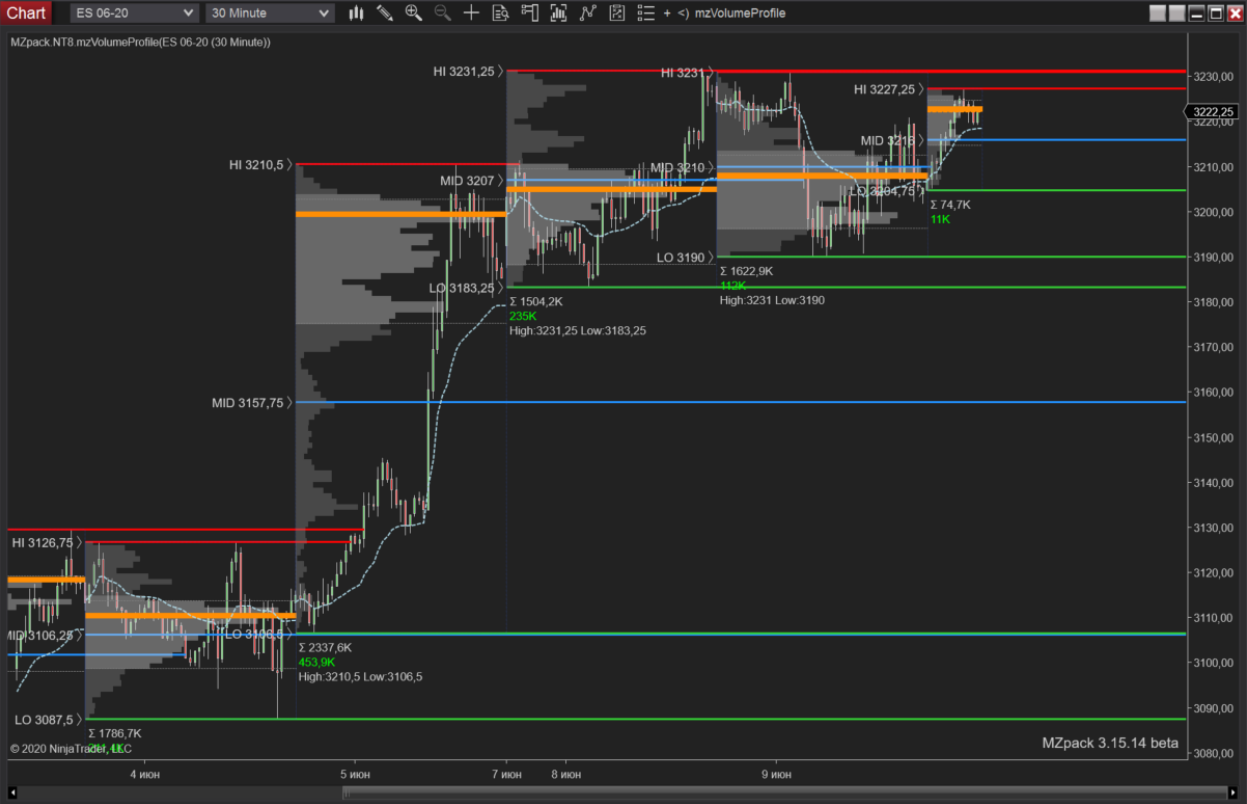
TPO levels in Custom profile mode. In **Custom** profile mode 'Attach to' value can be 'All' only, and the last added custom profile has T index = 0.
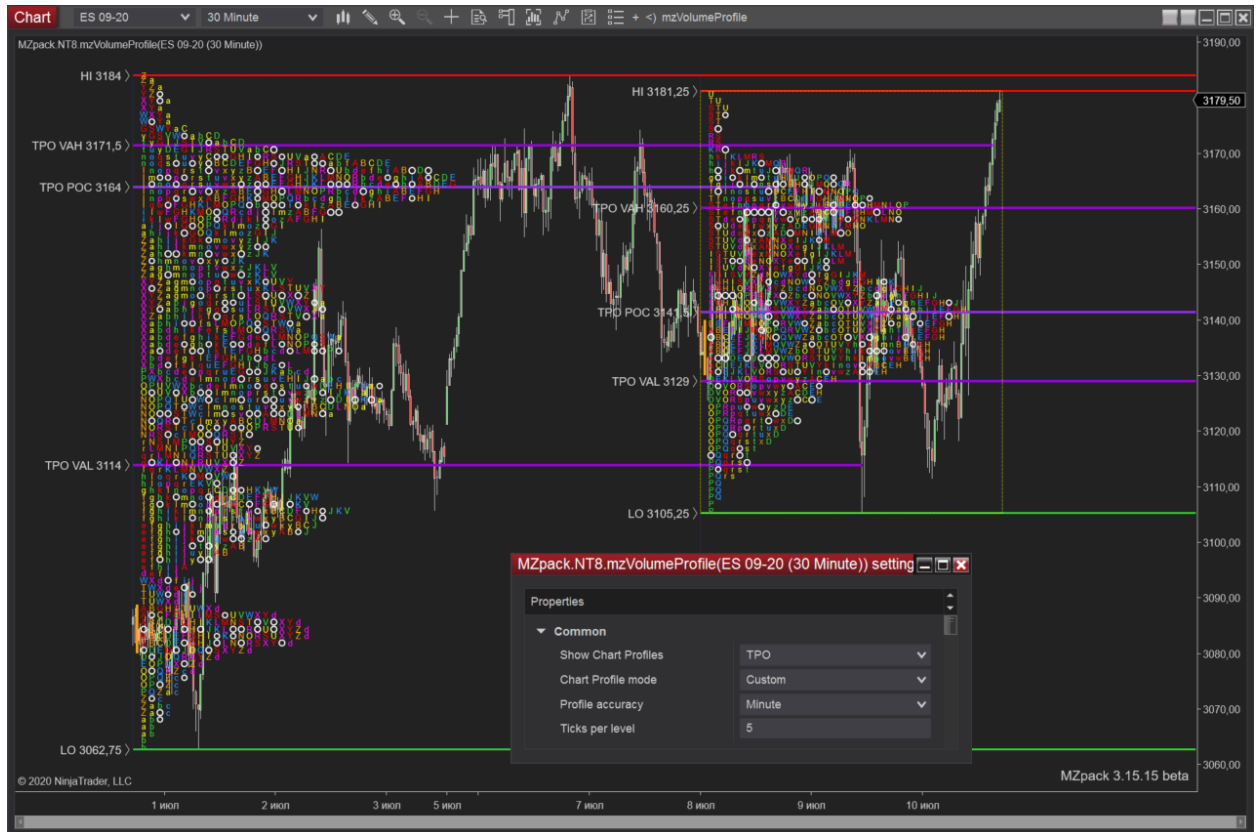
STACKED PROFILES

You can add up to three independent Volume Profiles at right margin of the chart. Profiles support cumulation and shifting functions.

To build **cumulated profile** choose one of the profile modes: Sessions, Bars, Days, Weeks, Months. To specify number of last profiles to be cumulated set 'Profile N last' parameter. Example. Sessions[2] builds volume profile for two last sessions.

To build **a profile for arbitrary period** choose one of the profile modes: Session, Bar, Daily, Weeklu, Monthly, Quarterly, Yearly.
Example. Daily[1] builds volume profile for one day ago. Monthly[0] builds volume profile for current month.

In Extended mode for POC/VAH/VAL levels those start at profile start on the chart and continue to correspondent stacked profile.
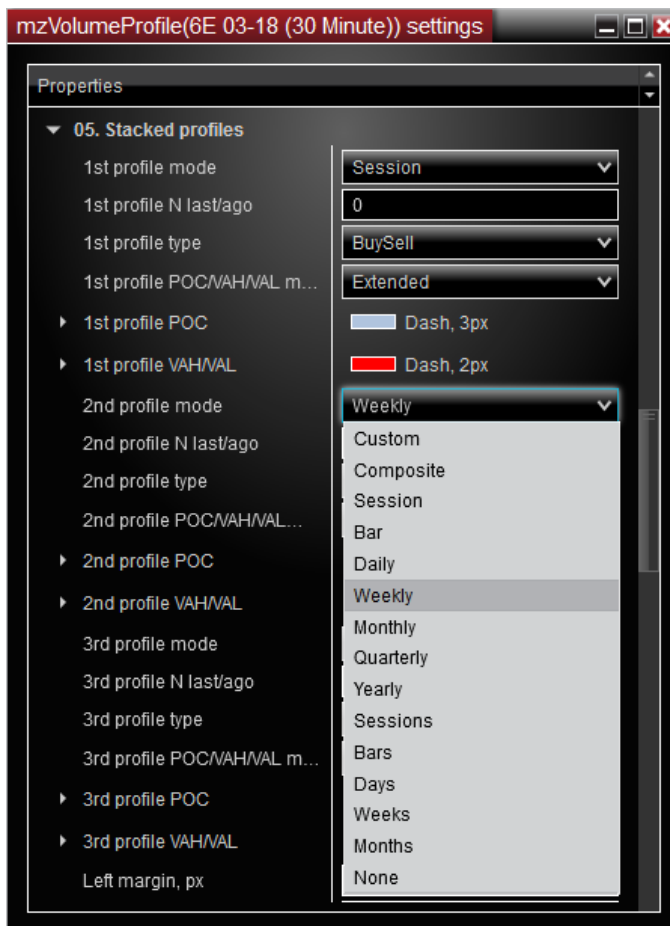
All profiles arranged automatically on chart right margin. Increase margin value in chart properties to free more space for Stacked profiles.

WORKING TIME

'General - Working time' option is used to define working time interval of the indicator. Set 'Start time' as beginning and 'Stop time' as the end of time interval when indicator will process market data. You can specify the time of day only but not the date.

In mzVolumeProfile set 'Chart Profile mode' = **Continuous** if **Working time** is enabled. In this mode, a new profile is created at the beginning of each time interval.
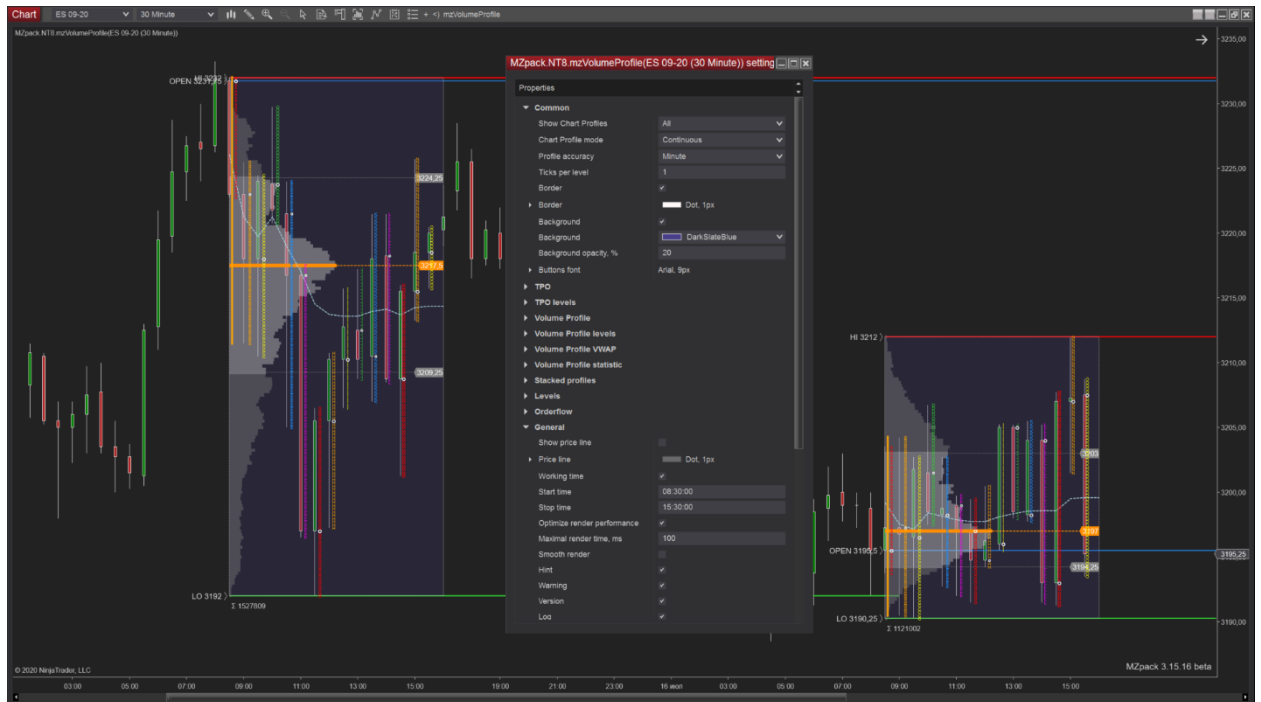
Bars closed *before* Start time will be *excluded* from processing. Bars opened *after* Stop time will be *excluded* from processing. If epy time of the bar falls into epy Working time interval (even partially), this whole bar will be processed.

Note, that time of the bar on the chart is its CLOSE time. Next images illustrate how Working time is applied to such bars markup when **Profile accuracy = Minute**.
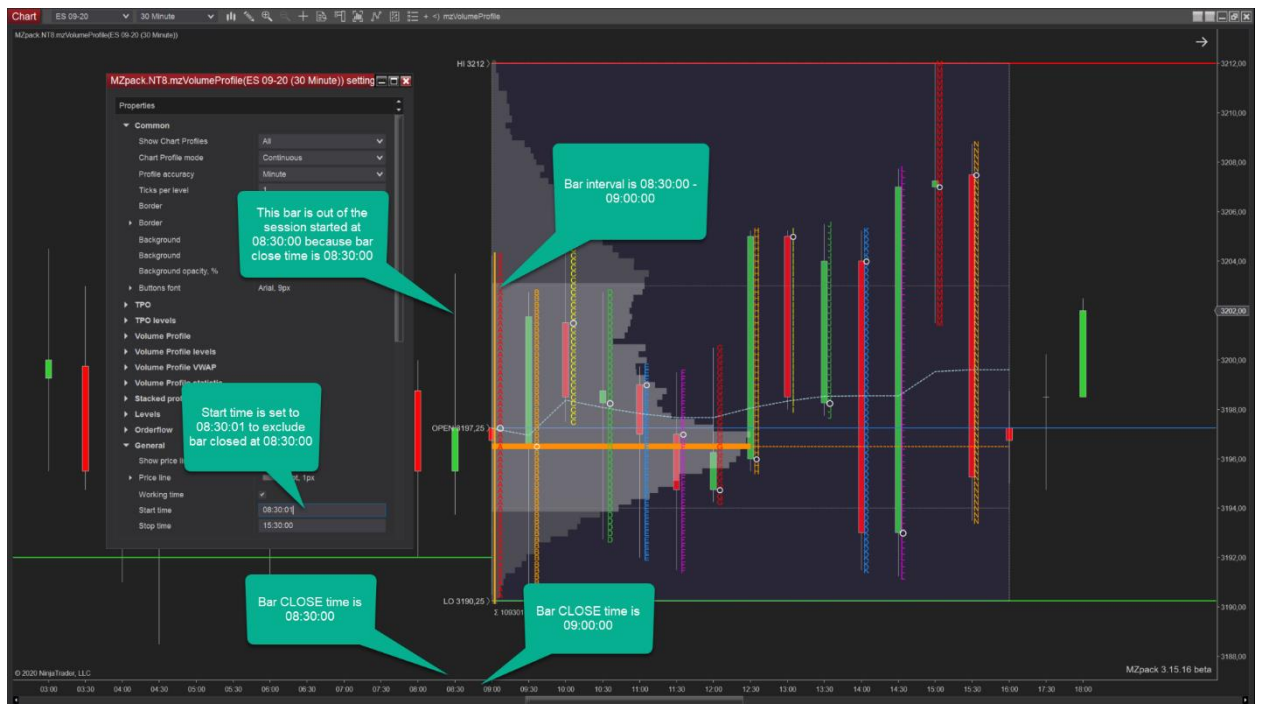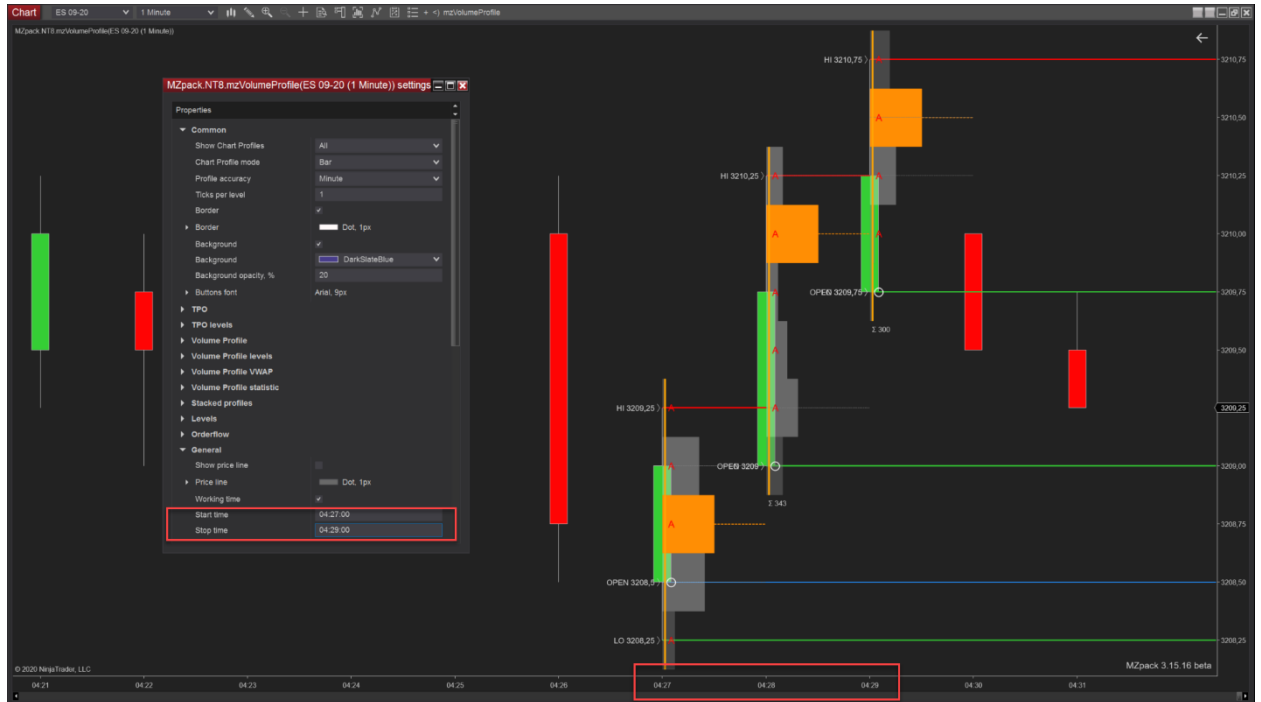
mzVolumeDelta Indicator

The mzVolumeDelta indicator combines Buy/Sell volume indicator and Delta indicator with iceberg search algorithms.

Indicator page: https://www.mzpack.pro/product/mzvolumedelta-for-ninjatrader-8/

Indicator settings

Common

- **Mode**. Indicator mode
  - a. Volume
  - b. Delta
- **Bar width**. Histogram or candle width
  - a. Auto
  - b. Custom
- **Bar width, px**. Bar width in pixel if Custom width selected
- **Zero line**. Show zero level line

Filters

- **Trade min.** Trades less than this value are not counted
- **Trade max.** Trades greater than this value are not counted. Use -1 for unlimited trade size
- **Display color**. Histogram bars and candles (in both Volume and Delta modes) are rendered with 'Below display color' color if its values are less than this value
- **Below display color**.

Volume mode

- **Show**
  - a. Volume
  - b. Icebergs. *Orderflow - Reconstruct tape* option must be enabled.
  - c. Both. *Orderflow - Reconstruct tape* option must be enabled.
- **Align**. Histogram align modes
  - a. Total
  - b. Stacked
  - c. Opposite
- **Iceberg algo**.
  - a. Hard
  - b. Soft

- **Volume alert.** Enable trade volume alert
- **Volume alert threshold.**
- **Volume alert sound.**
- **Iceberg alert.** Enable iceberg volume alert
- **Iceberg alert threshold.**
- **Iceberg alert sound.**

Delta mode

- **Show**
  - a. Histogram
  - b. Candles
  - c. Cumulative
- **Cumulate session only**. Reset cumulative delta on each new session
- **Bar Delta alert.** Enable alert on absolute delta of a bar
- **Bar Delta alert threshold.**
- **Bar Delta alert sound.**

Presentation

- **Bar color scale (Volume mode)**. Maximal value for color coding will be as selected
  - a. **Chart**. Color scale calculated among all bars visible on chart
  - b. **All**. Color scale calculated among all loaded bars
- **Color code (Volume mode).** UpDown, Delta. Set 'Color code' = Delta to code volume bars by delta value. This will help to highlight volume bars with big delta. Set 'Color code' = UpDown to code volume bar color by chart bar type - up or down.
- **Color mode**
  - a. Solid. Selected solid color for all data
  - b. Saturation. Saturation depending on volume/delta values
  - c. Heatmap. Heatmap colors by volume/delta values
  - d. GrayScaleHeatmap. Gay scaled heatmap colors by volume/delta values
  - e. Custom. 4 custom color to use depending on Volume or Delta value
- **Buy volume/Positive delta**. Color for Buy volumes
- **Sell volume/Negative delta**. Color for Sell volumes
- **Candle/Bar border**. Stroke for candles bodies border
- **Buy Iceberg (Volume mode)**.
- **Sell Iceberg (Volume mode)**.
- **Candle wick**. Stroke for candles wicks border
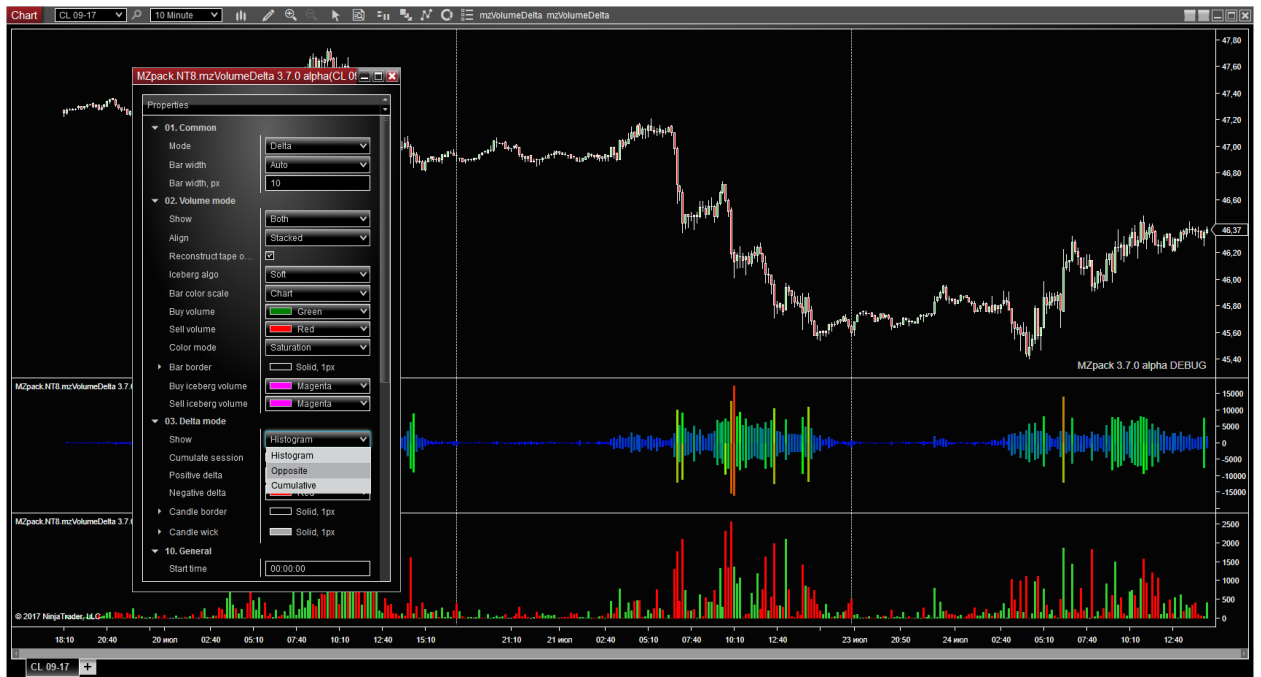
Reconstruct tape mode on 1 Tick chart
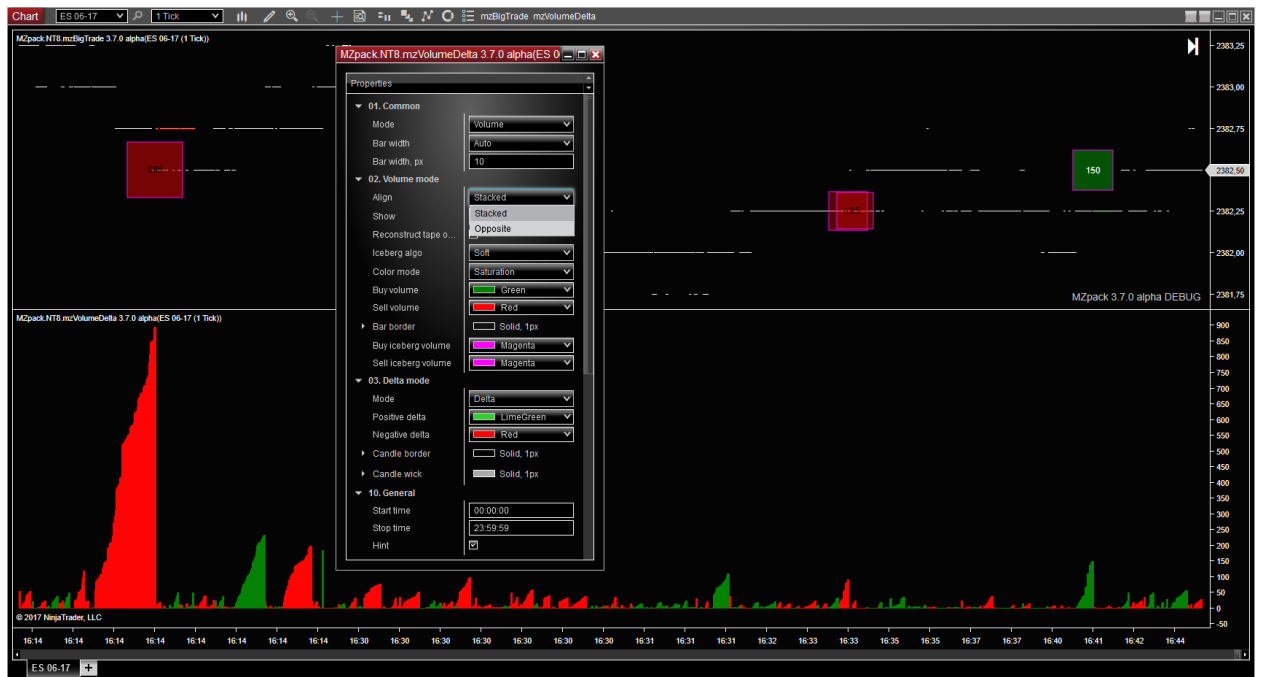
Cumulative delta mode

mzDeltaDivergence Indicator

The mzDeltaDivergence indicator shows Price/Delta divergences on the chart.

The indicator includes all the functionality of mzVolumeDelta indicator. For Volume/Delta features see mzVolumeDelta indicator topic, please

mzDeltaDivergence uses Zigzag indicator to spot breakpoints of a trend, and cumulative delta (by sessions) to find devergences. Here is an example of Price/Delta divergences.
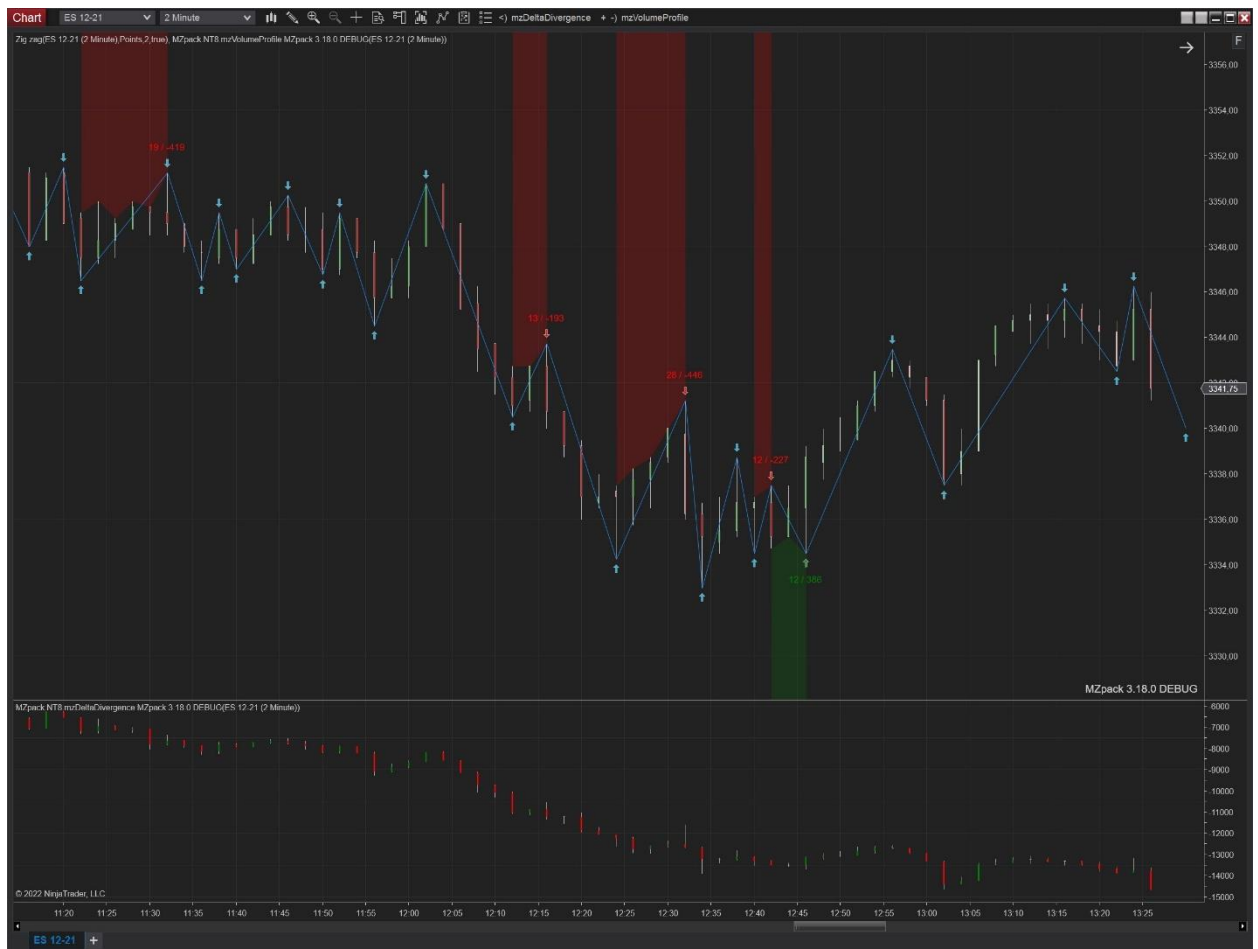
Indicator settings (Divergence category only! For Volume/Delta features see mzVolumeDelta indicator topic, please)

Divergence

- Show. Show Volume/Delta divergences on the chart
- Zigzag: deviation type. Value or Percent

- Zigzag: deviation threshold. The threshold as Value or Percent. Value is the deviation of the price in **ticks** (not points!). The deviation of the price in **percents** is calculated inside current session

- Zigzag: use High-Low

- Price deviation: type. Value or Percent. Value is the deviation of the price in ticks. The deviation of the price in **percents** is calculated inside current session.

- Price deviation: min

- Price deviation: max. Set -1 for unlimited value

- Delta deviation: type. Value or Percent. Value is the deviation of delta in contracts. The deviation of delta in percents is calculated inside current session.

- Delta deviation: min

- Delta deviation: max. Set -1 for unlimited value

- Divergence: deviation logic

  PriceAndDelta – Price and Delta both must have deviations

  PriceOrDelta – at least one of deviation must be met: Price or Delta

- Breakpoint: show. Show arrows near breakpoints

- Breakpoint: offset, px

- Breakpoint: color

- Statistics: show. Show stat values near a breakpoint

- Statistics: offset, px

- Statistics: font

mzMarketDepth Indicator

The mzMarketDepth indicator displays limit orders resting in the order book (or Depth Of Market) and shows its volumes on the chart as historical data, and as real-time data via colored histogram.

Indicator page: https://www.mzpack.pro/product/mzmarketdepth-for-ninjatrader-8/

Indicator settings

Common. Common settings for all presentations of DOM

- **Multiple Market Maker**. Market depth support for stock markets. Don't turn it on for futures via IQfeed!
- **Depth of market**. Given number of DOM levels will be displayed. *Actually, what you see in NinjaTrader Level II is not the whole order book but only its visible part. DOM is a frame into real market order book. Some markets provide full order book data (e.g. Crude oil), but e.g. E-mini S&P500 provides only 10 levels nearest to the current price.*
- **Bid**. Color for buy limit-orders
- **Offer**. Color for sell limit-orders
- **Extremal Bid**. Color for extremal buy limit-orders
- **Extremal Offer**. Color for extremal sell limit-orders
- **Bid text**. Color for bid values
- **Offer text**. Color for offer values
- **Extremal Bid text**. Color for extremal bid values
- **Extremal Offer text**. Color for extremal offer values

Realtime DOM. Real-time DOM on right chart margin

- **Show histogram**. Show DOM histogram or "ladders"
- **Histogram border**
- **Show volumes**. Show limit orders volume resting in DOM
- **Volumes font**
- **Code extremal**. Code extremal volumes by extremal color saturation
- **Color mode**. Solid, Saturation, Heatmap, GrayScaleHeatmap. Custom is not supported.
- **Saturation preset**. Saturation level if Color mode = Saturation
- **Imbalance: show.** Show imbalance in levels on Bid or Offer sides independently

- **Imbalance: ratio**. Ratio of the Imbalance. For example, if ratio is 2 then all volumes that are two times greater than average volume of given DOM side will be marked as imbalanced.
- **Imbalance: Bid color**. Color for the Imbalance on Bid side
- **Imbalance: Offer color**. Color for the Imbalance on Offer side
- **Cumulative: show**. Show two lines for volumes cumulated level by level on each DOM side.
- **Cumulative: Bid color**. Line for cumulative Bids
- **Cumulative: Offer color**. Line for cumulative Offers
- **Ladders: width, px**. Maximal histogram width in pixels
- **Ladders: extend.** All ladders have 'Ladders width' size.
- **Margin: left, px**
- **Margin, right px**
- **Margin: control right**. Control minimal width of right side margin of the chart

Historical DOM

- **Show**. Show historical DOM on the chart
- **History depth, bars**. Maximal number of chart bars for which historical DOM will be shown. History depth range is 10..5000. Use smaller values if you have performance issues.
- **Filtering mode**. Filtering mode applied to historical DOM
  a. **Absolute**. Use filter values in contracts
  b. **Percentage**. Use filter values in %
  c. **AdaptiveLess**. See detailed description further.
  d. **AdaptiveMore**. See detailed description further.
- **Display volume, %.** Display only limit-orders with given % of volumes. *E.g. if value is 60% and size of maximal limit-order in the order book is 100-Lot, then only 41-Lot limit-orders or greater will be displayed. If maximal limit-order will become 200-Lot, then only 81-Lot limit-orders or greater will be displayed*
- **Extreme volume, %.** Given % of volumes from bigger end *of displayed volumes* is treated as extremal
- **Display volume, contracts**. Display DOM levels with volume greater than or equal to this value
- **Extreme volume, contracts**. DOM levels with volume greater than or equal to this value treated as extremal.
- **Code extremal**. Code extremal volumes by extremal color saturation
- **Color mode**
  a. Solid. Selected solid color for all data
  b. Saturation. Saturation depending on volume/delta values
  c. Heatmap. Heatmap colors by volume/delta values

> d.   GrayScaleHeatmap. Gay scaled heatmap colors by volume/delta values
>
> e.   Custom. Not supported.

- **Saturation preset**. Saturation level if Color mode = Saturation.
- **Hold levels**. Keep displaying levels out of Depth of market scope. *Actually we don't know is still there some liquidity or not, but we can keep and display last quantities for analysis*
- **Extend levels, px.** Extend levels on right chart margin on given number of pixels
- **Hold higher volume**. Keep displaying maximal limit-order volume inside current bar on a price level. *You can spot liquidity spikes by this way.*
- **Show popup info**
- **Show volumes**
- **Show max volumes.** Show maximal volume in brick in brackets next to realtime volume
- **Volume font**

Liquidity

- **Show overall**. Show overall (total) liquidity plots at the bottom/top of the chart
- **Overall type**.
  - a.   *OfferBid*. Displays total volume on Offer and total volume on Bid (two lines)
  - b.   *OfferBidDelta*. Displays total volume on Offer **minus** total volume on Bid (one line or candles)
  - c.   *BidOfferDelta*. Displays total volume on Bid **minus** total volume on Offer (one line or candles)
- **Plot**. *Line* or *Candle. Candle* plot is for *OfferBidDelta/BidOfferDelta* types.
- **Plot Bid/Up-candle**. Line color/style for liquidity on Bid and color for Up-candles
- **Plot Offer/Down-candle**. Line color/style for offer liquidity and color for Down-candles
- **Plots position**. Plots' position on the chart: Bottom or Top
- **Plots height, px**. Height of plots area in pixels.
- **Plots background.** Show plots background.
- **Plots background color.**
- **Plots background opacity, %.** 0..100

Liquidity Migration

- **Show**. Show liquidity Migration
- **Added volume filter**. Mark limit-orders being added to order book with total size greater or equal to this value

- **Removed volume filter**. Mark limit-orders being removed from order book with total size greater or equal to this value
- **Added offer**. Color for 'added liquidity' mark on offer
- **Added bid.** Color for 'added liquidity' mark on bid
- **Removed offer**. Color for 'removed liquidity' mark on offer
- **Removed bid**. Color for 'removed liquidity' mark on bid
- **Show overall**. Show overall migration plots at the bottom/top of the chart
- **Overall type**.
  a. *OfferBid*. Displays total for migration on Offer and total for migration on Bid (two lines)
  b. *OfferBidDelta*. Displays total for migration on Offer **minus** total for migration on Bid (one line or candles)
  c. *BidOfferDelta*. Displays total for migration on Bid **minus** total for migration on Offer (one line or candles)
- **Cumulate**. Cumulate migrations by each side of the DOM from bar to bar. Migration for a new bar will start from migration of previous bar.
- **Plot**. *Line* or *Candle*. *Candle* plot is for *OfferBidDelta/BidOfferDelta* types.
- **Plot Bid/Up-candle**. Line style for bid migrations or color for up-candles
- **Plot Offer/Down-candle**. Line style for offer migrations or color for down-candles
- **Plots position**. Plots' position on the chart: Bottom or Top
- **Plots height, px**. Height of plots area in pixels.
- **Plots background.** Show plots background.
- **Plots background color.**
- **Plots background opacity, %.** 0..100

Imbalance. Order book imbalance calculated as percentage ratio of buy and sell limit-orders total quantities or volumes

- **Show**. Show order book imbalance
- **Imbalance, %.** Threshold value to display imbalance. Note that changing this value does not affect on historical bars
- **Bid imbalance**. Line for imbalance on bid
- **Offer imbalance**. Line for imbalance on offer
- **No imbalance**. Line for bid-offer balance

Notifications

- Extreme alert. Alert on extremal limit-order volume
- Extreme sound

If **Imbalance – Show** setting is checked, mzMarketDepth indicator auto-scales chart to fit DOM on the chart. If you don't want this uncheck **Auto scale** option in Visual section from NT8 built-in settings window.
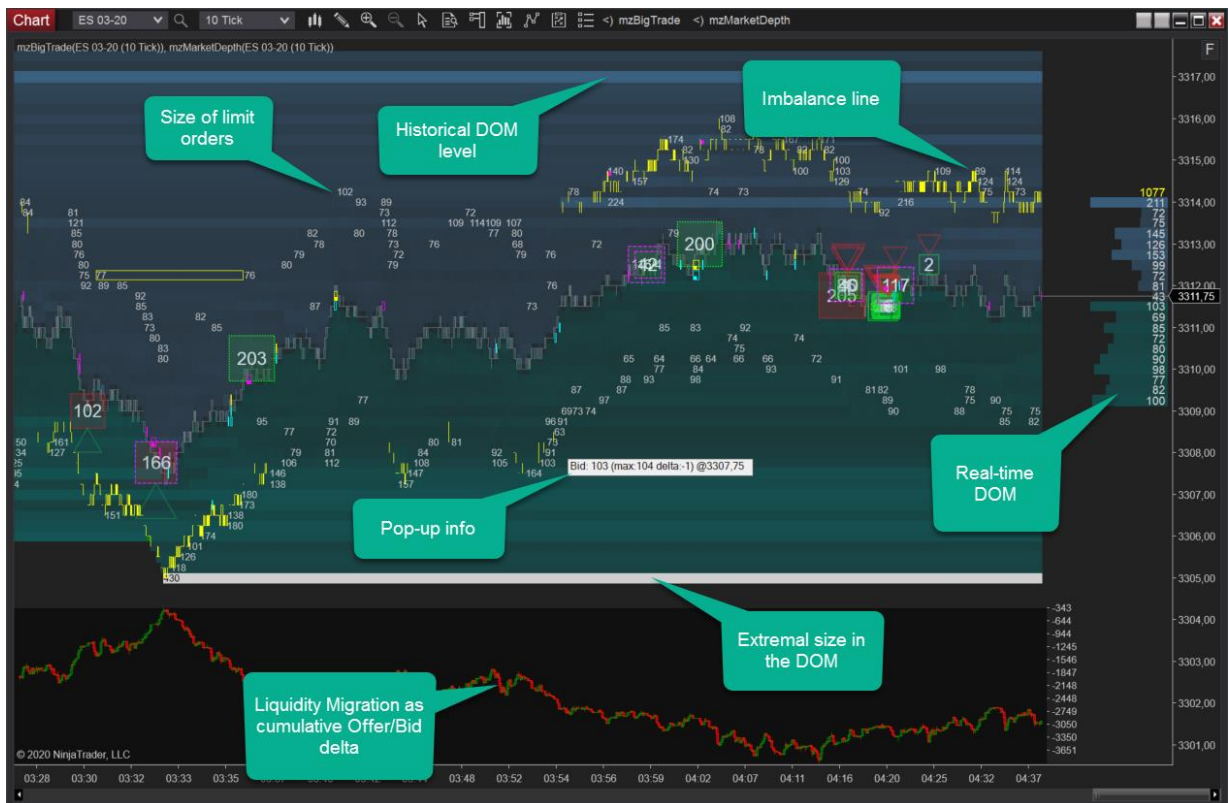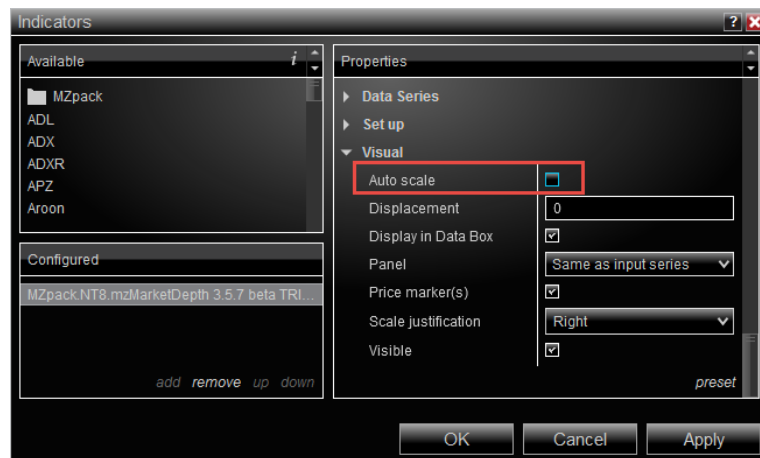
ADAPTIVE FILTERING MODES

Adaptive filtering modes use initial percent as a starting point for filtering.

AdaptiveMore calculates minimal DOM order size to display starting from the point when DOM was initialized. Later the percent will be changed automatically to maintain minimal DOM order size for displaying.

AdaptiveLess (default value) mode is similar to AdaptiveMore except that minimal DOM order size to display will be slightly adjusted in time.

Both modes allow seeing clear historical DOM picture automatically adapting to future orders' sizes.
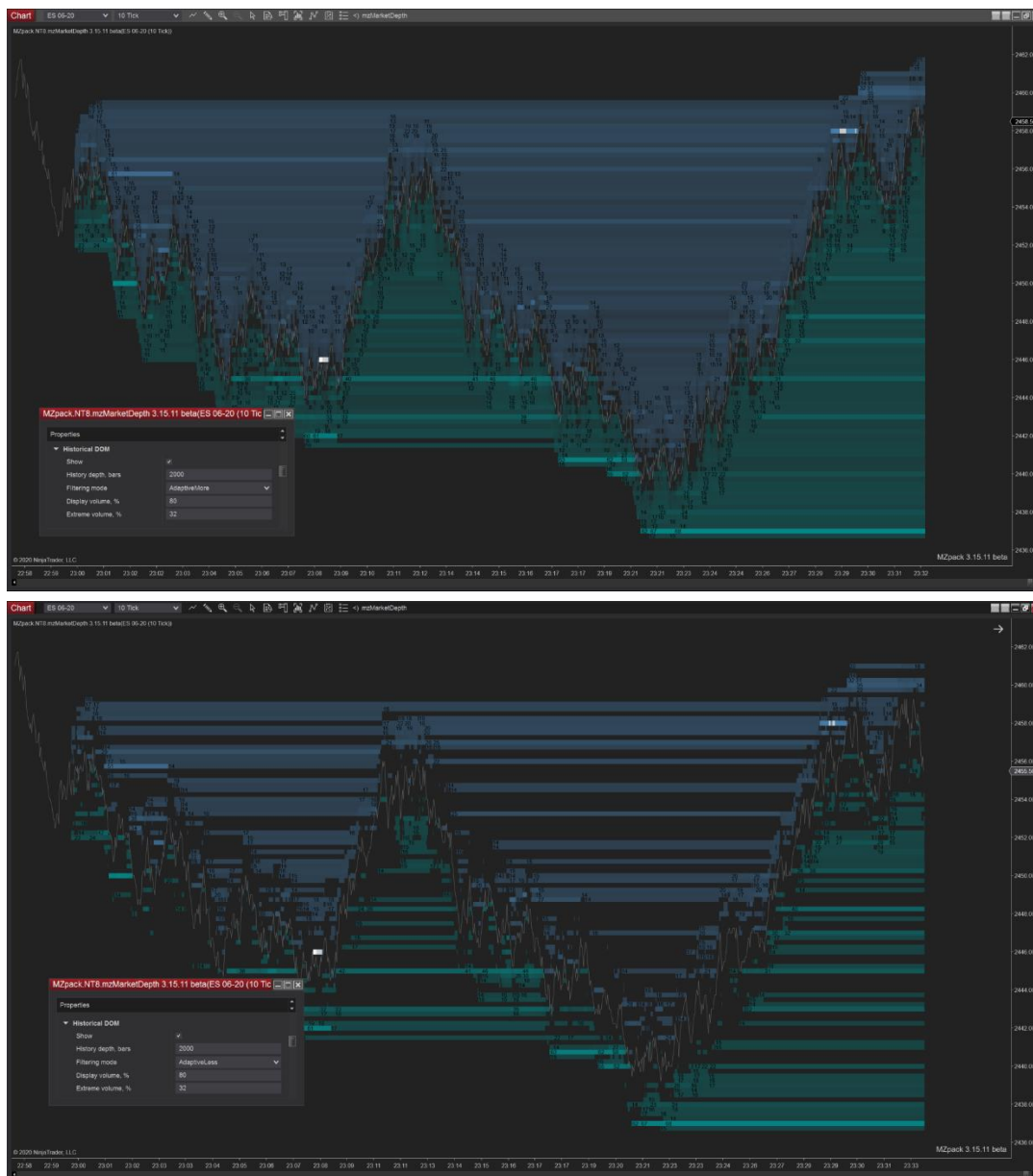
LIQUIDITY

Liquidity plots are available from Liquidity category of the settings.

Supported plots:

- **OfferBid.** Plots total volume on Offer side and total volume on Bid side of the DOM (two lines)
- **OfferBidDelta.** Plots total volume on Offer side minus total volume on Bid side of the DOM (one line or candles)
- **BidOfferDelta.** Plots total volume on Bid side minus total volume on Offer side of the DOM (one line or candles)

As an example, see Liquidity Cross pattern based on OfferBid type. A symmetrical cross can be drawn on OfferBid plots. The entry can be made after the right parts of the cross break its low and high bound.
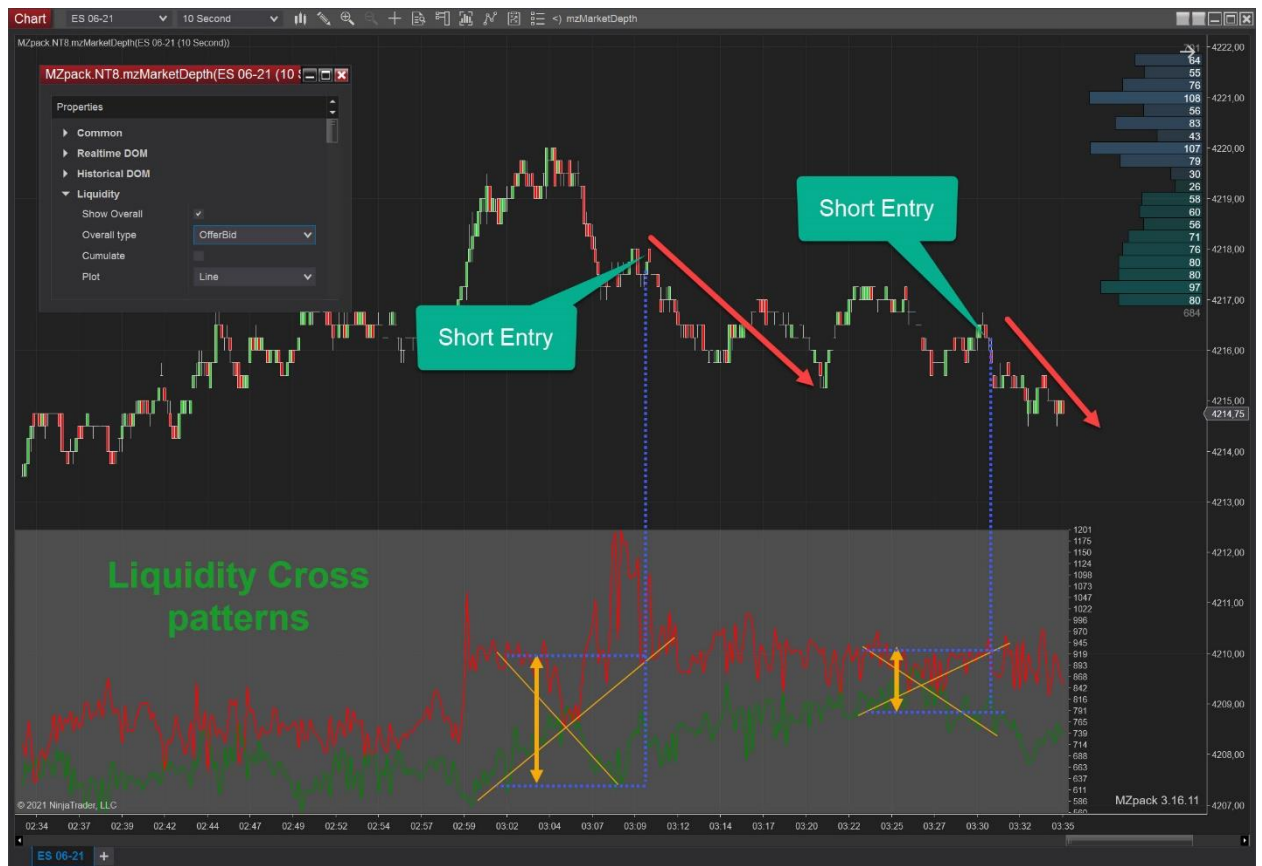
LIQUIDITY MIGRATION

Liquidity Migration is the process of adding and removing limit orders in DOM. We can spot migration per each DOM level.
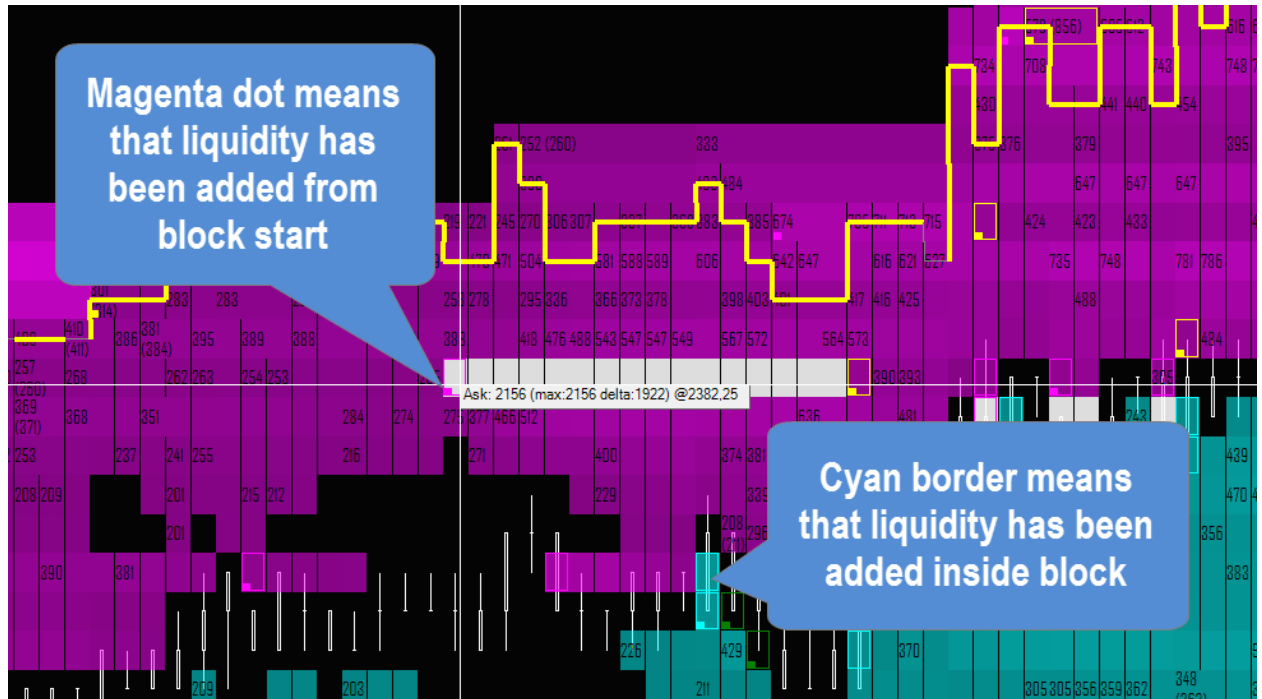
Also, you can see total migrations on each side of the DOM and delta of totals in lines and candles presentations at the bottom of the chart.
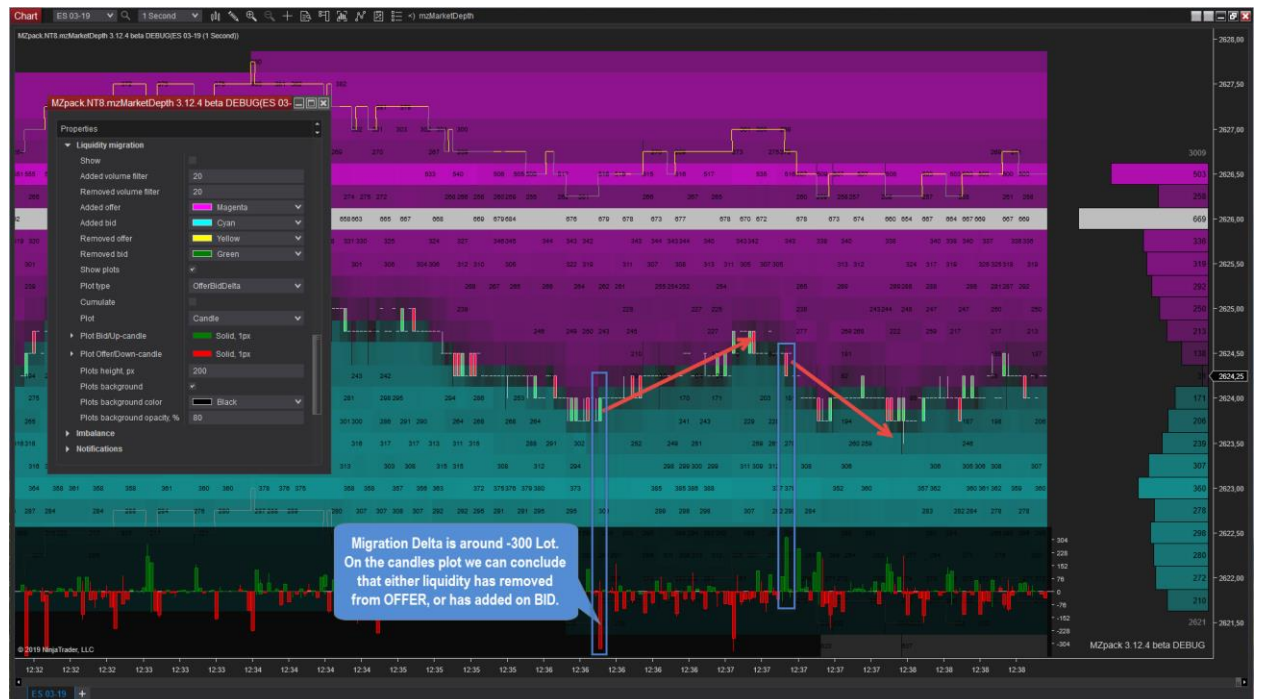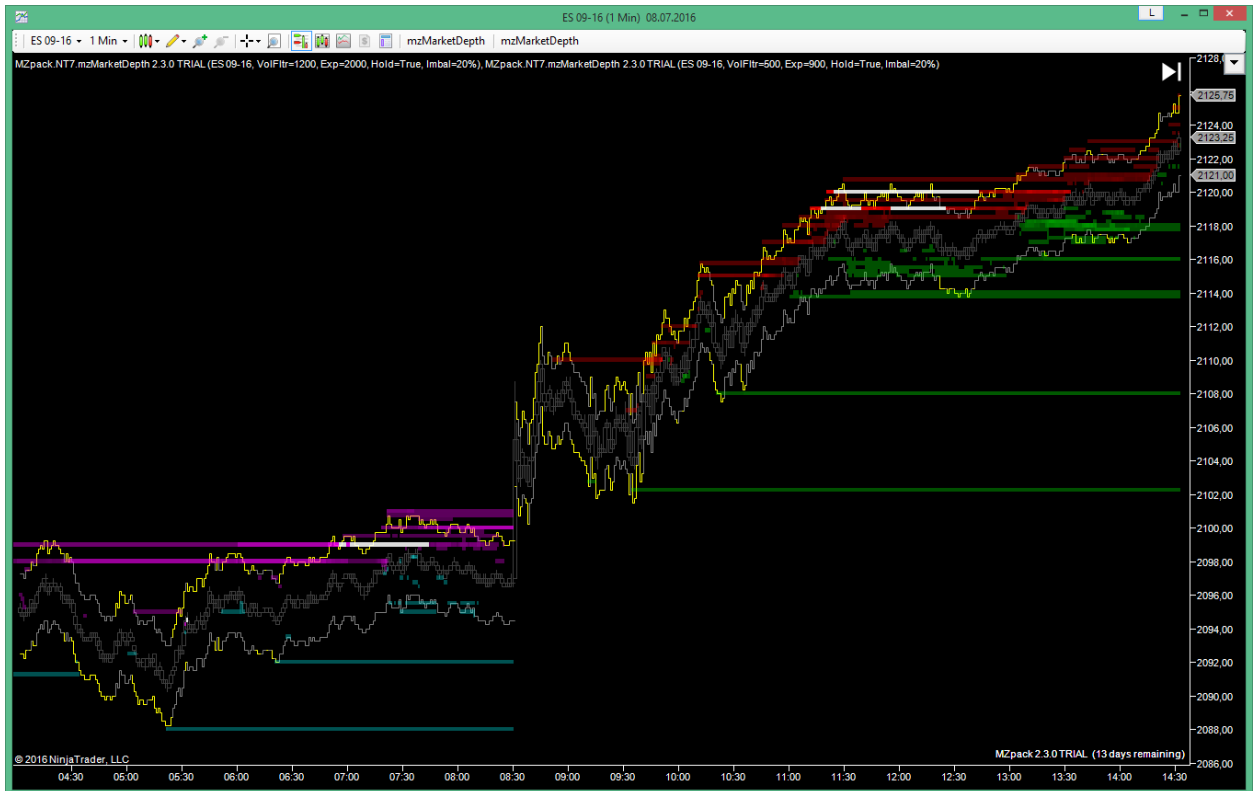
On this ES chart two mzMarketDepth indicators with different filters have been placed. One indicator is for Europe session and another one for American session. Time on chart is UTEC-5 (EST USA & Canada). See **Indicators' general parameters** section to learn how to use Strat/Stop time option.

Levels Feature

Levels. Default settings for new alert levels added to the chart

- **Enable.** Enable levels functionality
- **Support level**. Default line style for support level (level is below the price)
- **Resistance level**. Default line style for resistance level (level is above the price)
- **Value position.** Left, Right, AboveLeft, AboveRight, None
- **Value/Label color.** Color for level label and price
- **Value/Label font.** Default font for level label and price
- **Alert**. Alert on conditions
- **Aler on**. Default condition for alert
  a. LevelCross
  b. LevelTouch
- **Rearm**. Rearm alert after it has been fired
- **Rearm interval, sec**. Alert will be rearmed after this period of time. Set to zero to rearm alert immediately after it has been fired
- **Sound**. Alert sound
- **Add/remove: Mouse Left +.** Keyboard key to add or remove a level
- **Modify: Mouse Left +.** Keyboard key to modify a level
- **Drag with Mouse.** Drag level with Mouse left click

Default mouse + key combinations to set and manage alert levels

- Add a level: Left Mouse Click + Left Shift key
- Remove a level: Click remove button or Left Mouse Click + Left Shift key on a level
- Adjust level duration: Left Mouse Click + Left Alt key on a level

To drag a level click mouse left button (don't hold) on the level and move the cursor. Then click again.

To open level properties click left mouse button on properties button.

Levels are saved in workspace xml file when you click Save workspace. You can export levels or replicate those on another chart using Templates – Save as… command from context menu of the chart.

NOTE. Levels not saved in workspace file will not be restored after reloading indicator or re-establishing data connection. Save your workspace after adding/removing levels.
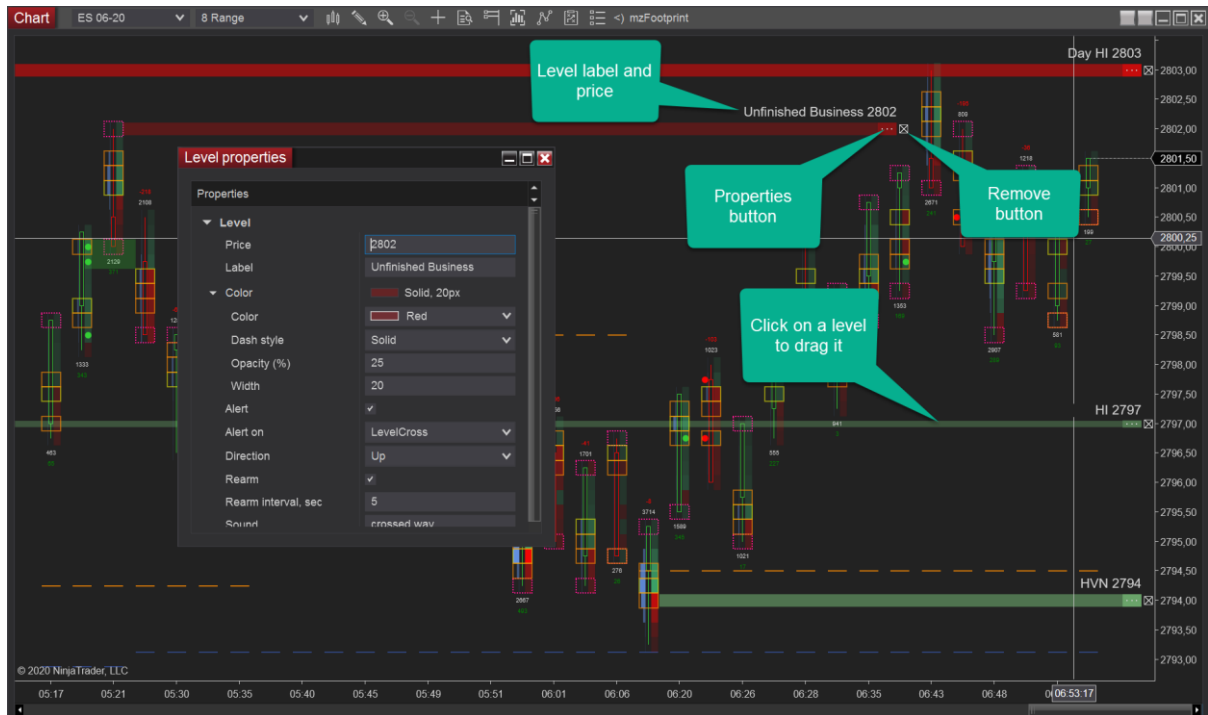
When adding a level click inside Footprint bar or Volume profile/TPO to set this visual as a sarting point of the level. See image.
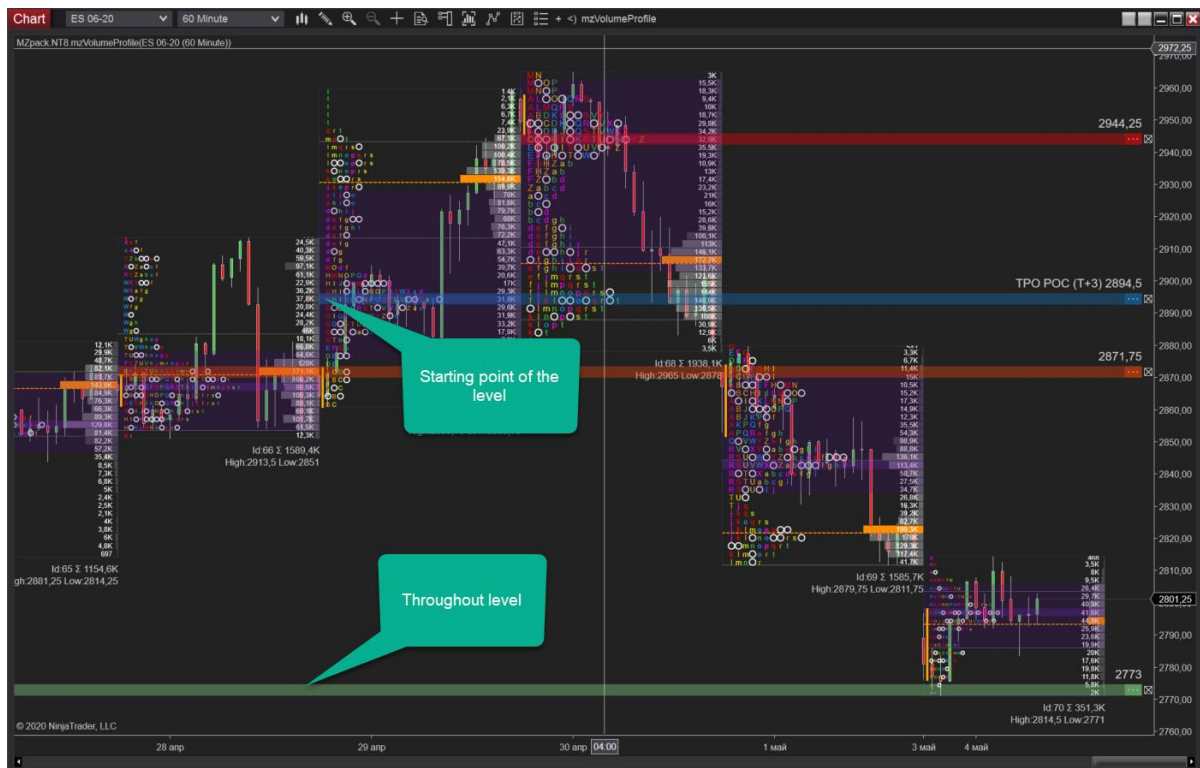
Orderflow Settings

This group of settings is available for: mzBigTrade, mzFootprint, mzVolumeProfile, mzVolumeDelta indicators.

Orderflow

- **Calculation mode**. **BidAsk** – the most accurate mode of calculation of buy/sell volumes. Requires historical Bid-Ask data; **UpDownTick** – use this mode for if your data feed doesn't provide historical Bid-Ask data (e.g. Forex market). All up-ticks and subsequent ticks are buy volumes, all down-ticks and subsequent ticks are sell volumes. The **Hybrid** mode uses UpDownTick calculation for historical data and BidAsk calculation for real-time data, which is much accurate. [Read more](#) on how to use it eg. in Indian Market when you don't have historical bid/ask data.

- **Spread trades**. How to process trtades insude current spread. **Split** – 50/50 for buy and sell sides, **LastKnowSide** – add trade volume to the side of the prior trade, **Ignore** – ignore trades inside spread (default)

- **Non-Crypto volumes divider.** Trade volumes divider for non-cryptocurrency instruments. It's useful for NSE instruments (eg. NIFTY). Note, that Crypto volumes are being converted automatically for instruments with CryptoCurrency type in the instrument settings (eg. BTCUSD). See also 'General - Crypto decimal places'

- **Reconstruct tape**. Reconstruct tape using time stamps and Level 1 events.

- **'Reconstruct tape' apply.** mzBigTrade indicator only. **ChartReload** – reload the chart to apply new value of Reconstruct tape option (default, use it to minimize memory consuption); **OnTheFly** – apply value of Reconstruct tape option on-the-fly (more memory required).

Note! Using On-the-fly method or disabling 'Reconstruct tape' will significantly increase memory consumption.
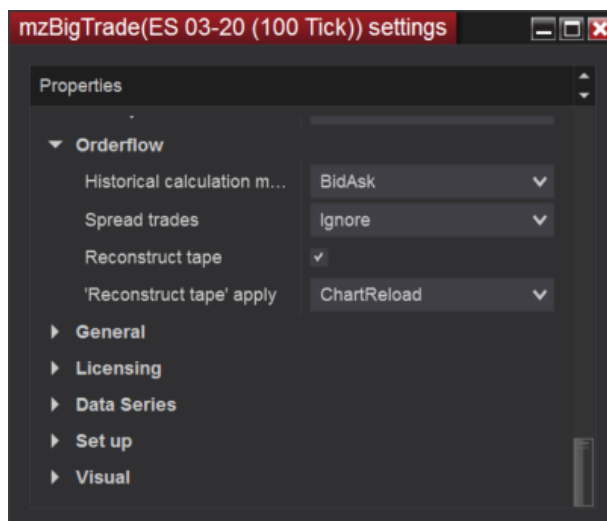
General Settings

General

- **Show price line**
- **Price line**. Stroke for price line
- **Crypto decimal places**. Decimal places for volumes of crypto instruments: 1..8
- **Working time**. Use Start/Stop time parameters as working time interval
- **Start time**. Indicator start time. Time is bar close time. Bars closed *before* Start time will be *excluded* from processing.
- **Stop time**. Indicator stop time. Time is bar close time. Bars closed *after* Stop time will be *excluded* from processing.
- **Optimize render performance.** Optional. Limit rendering time to free CPU/GPU resources and make chart responsive. If enabled, rendering time will not exceed '**Maximal render time, ms**'. Chart may 'flash' with this option. Adjust **Maximal render time** parameter to get balance between performance and visual comfort.
- **Maximal render time, ms**.
- **Smooth render**. Makes the picture smooth using Direct3D Antialias mode.
- **Hint**. Display indicator's hint on the chart if any.
- **Warning**. Display MZpack warning text on the chart.
- **Version**. Display MZpack version information on the chart. Requires NinjaScript reloading.
- **Log**. Log errors in trace files Documents\NinjaTrader 8\trace\.

Offten we need to set different settings for different trade sessions for an indicator. To do that, please:

1. Place, for example, two instances of the same MZpack indicator on a chart
2. Set desired parameters for each of them

3. Set Start and Stop time according to trade sessions (eg. Europe and American) for each of them

As a result one indicator will work eg. on Europe session only and the other one on American session only. See mzVolumeProfile indicator for more info.

By default Working time is off.
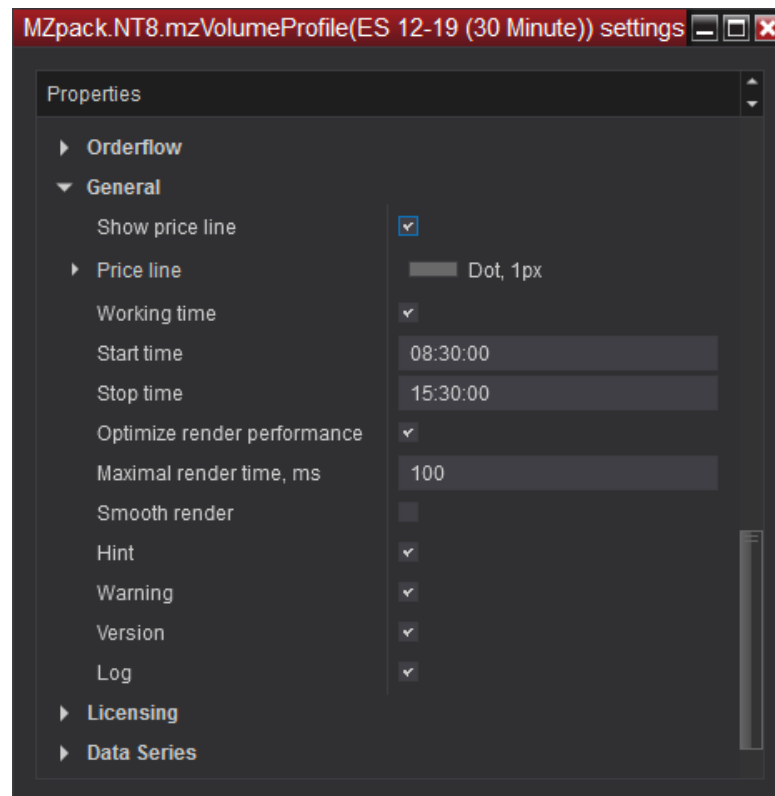
Sound Files for Alerts

List of sound files placed by setup in NinjaTrader sounds folder. To use sound for alert type file name with extension in the Sound parameter of an indicator.

- 6B.wav
- 6B_iceberg.wav
- 6B_strong_buy.wav
- 6C_iceberg.wav
- 6E_iceberg.wav
- buy.wav
- CL_iceberg.wav
- iceberg.wav
- make_money.wav
- ML.wav
- new_bigtrade.wav
- new_volume.wav
- pushing_ask.wav
- pushing_bid.wav
- sell.wav
- SL.wav
- strong_buy.wav
- strong_sell.wav
- sweep.wav

Managing Email Notifications

Enabling Notifications - Send Email in the settings of an indicator is not enough to get emails on alerts. You need to set up Email share service also.

In Ninja menu click Tools - Options. In General click Share services to add Email share service.

Watch this video on how to add Gmail account: https://www.screencast.com/t/2VshigfRtcVF

Managing Share Services: https://ninjatrader.com/support/helpGuides/nt8/?general_section.htm
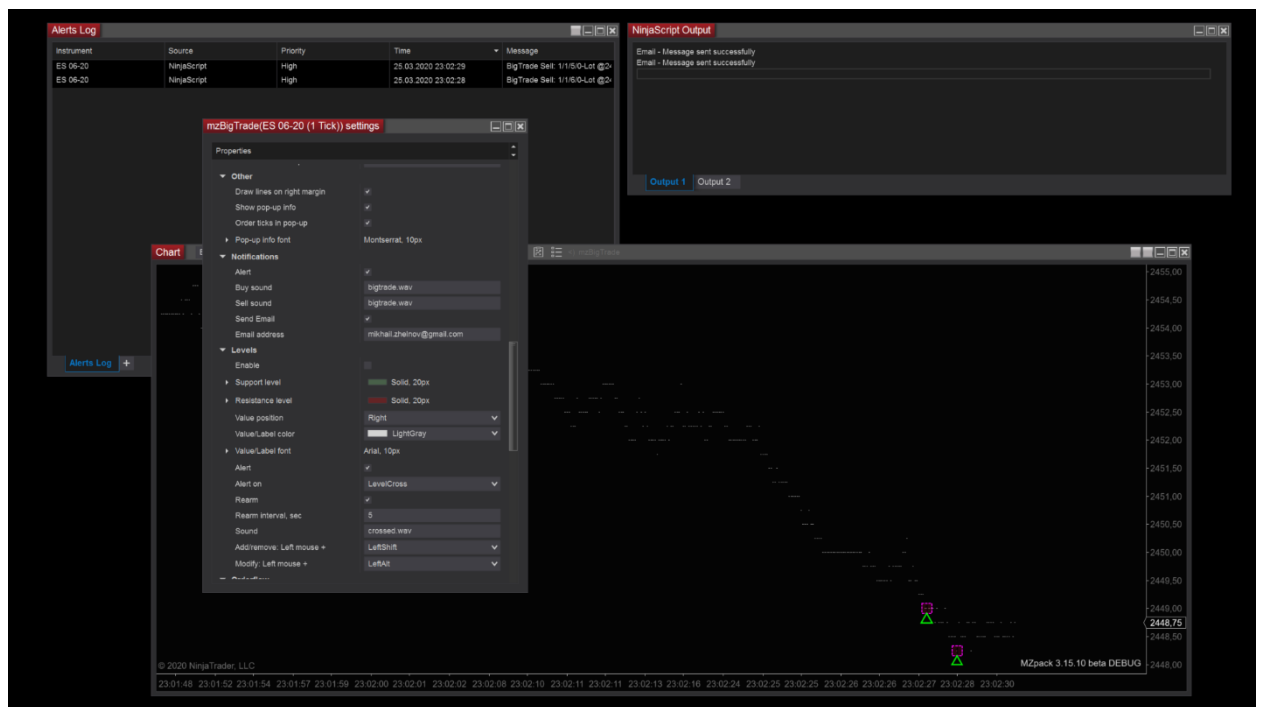
MZpack 3 API

INTRODUCTION TO MZPACK 3 API

MZpack 3 API is an interface that introduces traders to a programmable way of creating their own trading algorithms and automated strategies in NinjaTrader 8.

With an application interface, traders can use provided classes as building blocks to construct strategies with the minimum of coding. At the same time, MZpack 3 API grants granular flexibility in design instead of offering pre-made strategies.

What once took days or weeks, now takes hours or even minutes.

Please note that there are no indicators in the API that can be placed on the chart from the user interface.

MZpack 3 API for NinjaTrader 8 Online Help

BASIC CONCEPTS

Before beginning to design and develop your trading algorithms, you need to be aware of some basic concepts about algo design in MZpack API.

Pattern Concept

Formally defined, a **pattern** is a set of relationships that are satisfied by observations of a system or a collection of systems. A simple type of pattern is a repeating structure in space, like the patterns often found on wallpaper or tiles.

In API context, pattern is a set of **signals** and **filters.** A pattern can be validated by validating its signals and filters. Validated pattern has a **determined direction** which is LONG or SHORT.

Localization Of a Pattern

Normally, we are taking a trading decision when some conditions are met at some point in time considering the prior context. In MZpack API the decider is a pattern and it must be localized.

The criteria for the localization of a pattern are the price range and time period. The time period is substituted by a range of bars because often this is more suitable for order flow trading.

Signals and Filters. Logical Trees

Signal/filter is an entity that processes incoming market data and changes its **state** that is a set of properties: direction, price, time, etc.

Signals and filters are grouped into **logical trees**: signals tree and filters tree. Any tree is a logical AND/OR **desicion tree**. A tree contains two types of **nodes**: logical AND/OR and signal/filter node.

A pattern must contain at least one signal entity or node in signals tree. **Root node** is a logical AND/OR node that is created within the API. Note that a logical node cannot be a **terminal node**. The root node of the filters tree can be a terminal node, which means **no filters** present. Signal/filter node must be a terminal node.

The **structure** of each tree is checked upon initialization of the pattern. You will get `DecisionTreeInvalidStructureException` in case of errors.

If a range (localization) was specified for the tree, the tree must be validated in the given range. The signals tree must be validated **first**. Next, the filters tree must be validated starting with the bar at which the signals tree was validated. A price range can be applied for both trees also. Note that price ranges of trees are not related.

There are two types of patterns: Entry pattern and Exit pattern. Exit pattern is optional.

Entry pattern, once validated, must provide an **entry price. At least one entry price is required.** Any non-zero entry price generated by the filters tree will become the price for opening a position. If the filters tree entry price is zero, an entry price from the signals tree will be used to open a position.

Example of the signals tree. See image.
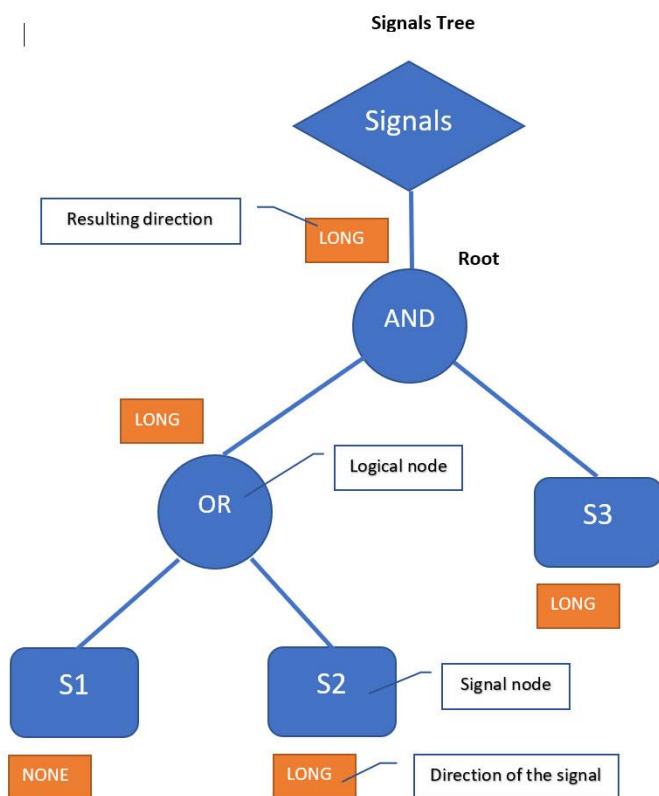
Signals Tree

This particular logical tree can be written as expression (S1 OR S2) AND S3.

All directions of signals must be reconciled with directions of other signals. Eg. in this example

(S1 => None OR S2 => Long) => Long

The resulting direction must be reconciled with the **allowd** direction of the whole pattern.

By default, **a short-circuit evaluation** used when evaluating logical **AND** nodes. In the common case, non short-circuit evaluation makes your strategy not dependent on the **order** in which signals appear on the chart. This is not true if pattern localization is used. For ranged patterns, the strategy will show different behavior depending on the order of signals under **AND** nodes in the trees even non short-circuit evaluation is used**.**

See constructor of `Pattern` class.

[Read more about short-circuit evaluation.](#)

BUILDING A SIMPLE STRATEGY

To illustrate these concepts, let us look at creating a simple strategy. Note that the API is used solely for accessing indicator data. For an advanced example, please see the next section.

MZpackStrategyBase is a base abstract class for any strategy based on MZpack API. Inherit custom strategy class from MZpack base strategy class. Then add private class members for any required MZpack indicators. These classes represent a set of MZpack indicators: StrategyFootprintIndicator, StrategyVolumeProfileIndicator, StrategyBigTradeIndicator, StrategyVolumeDeltaIndicator, StrategyMarketDepthIndicator.

```csharp
public class MyMZpackStrategy : MZpackStrategyBase
{
    StrategyVolumeProfileIndicator myVPIndicator;
}
```

In the constructor assign your delegates for creating desired indicators and market event handlers.

```csharp
public MyMZpackStrategy() : base()
{
    // Set OnCreateIndicators delegate
    OnCreateIndicators = new OnCreateIndicatorsDelegate(CreateMyIndicators);

    // Set OnBarCloseHandler for on bar close strategy
    OnBarCloseHandler = new OnTickDelegate(MyStrategyOnBarCloseHandler);
}
```

Implement handlers.

```csharp
protected List<TickIndicator> CreateMyIndicators()
{
    // Initialize new indicators list
    List<TickIndicator> indicators = new List<TickIndicator>();

    // Create CustomVolumeProfileIndicator instance with default settings
    // See CustomVolumeProfileIndicator class
    myVPIndicator = new StrategyVolumeProfileIndicator(this, @"Session Volume
Profile")
    {
        // Override defaults if required
        ProfileCreation = ProfileCreation.Session,
        ShowProfileType = ProfileType.VP  // Volume profile only
    };
    // Add indicator to the list
    indicators.Add(myVPIndicator);

    return indicators;
}
protected void MyStrategyOnBarCloseHandler(MarketDataEventArgs e, int
currentBarIdx)
{
```

```
            // Place logic of your strategy here accessing indicators' values.
            IVolumeProfile profile = (myVPIndicator.Profiles as
Model).GetItemByBarIdx(currentBarIdx) as IVolumeProfile;
            if (profile != null) Print(myVPIndicator.Volume);

            // Use managed approach for orders handling, ie. SetProfitTarget(),
            // SetStopLoss (), EnterShort(), etc.
        }
```

Examples of some implementations of handlers are available in samples provided. See THE
SAMPLES topic, please.

BUILDING AN ADVANCED STRATEGY

An advanced MZpack strategy is based on the pattern concept which also includes a proprietary ATM classes (`Entry`, `Trail`), logging feature, etc.

In the constructor of the strategy, assign `OnCreateAlgoStrategy` delegate.

```csharp
public MyAdvancedMZpackStrategy() : base()
{
    // Create MZpack algo strategy object to use built-in pattern and position management
    OnCreateAlgoStrategy = new OnCreateAlgoStrategyDelegate(CreateAlgoStrategy);

    // Set OnCreateIndicators delegate
    OnCreateIndicators = new OnCreateIndicatorsDelegate(CreateIndicators);
}
```

Implement `OnCreateAlgoStrategy` handler and return a new `MZpack.NT8.Algo.Strategy` instance with the desired settings.

```csharp
protected MZpack.NT8.Algo.Strategy CreateAlgoStrategy()
{
    return new MZpack.NT8.Algo.Strategy(@"My MZpack Algo Strategy", this)
    {
        OppositePatternAction = OppositePatternAction.Reverse, // Reverse position on validated opposite pattern
        LogLevel = LogLevel.ALL, // Log all events
        LogTarget = LogTarget.NinjaScriptOutput // Log to NinjaScript output only
    };
}
```

Create the required objects: pattern, signals/filters trees, entries, and initialize the strategy in `OnStateChange()`.

```csharp
if (State == State.Configure)
{
    EntriesPerDirection = 1;

    // Create entries
    Entry[] entries = new Entry[EntriesPerDirection];
    entries[0] = new Entry(Strategy)
    {
        EntryMethod = EntryMethod.Market,
        Quantity = Position_Quantity,
        SignalName = SIGNAL_NAME,
        StopLossTicks = Position_StopLoss,
        ProfitTargetTicks = Position_ProfitTarget,
        IsBreakEven = Position_IsBE,
        BreakEvenAfterTicks = Position_BEAfterTicks,
        BreakEvenShiftTicks = Position_BEShiftTicks,
        Trail = Position_IsTrail ? new Trail(Position_TrailAfter,
Position_TrailDistance, Position_TrailStep) : null
    };

    // Pattern
    Pattern pattern = new Pattern(Strategy,
        Logic.And, // Logic for root node of signals tree
```

```
                              new Range() { Bars = BigTrade_ClusterBarRange + 3, Ticks =
BigTrade_ClusterPriceRangeTicks + 5 },  // Range for signals (0 = no range limit)
                          Logic.And, // Logic for root node of filters tree
                          new Range() { Bars = 5 }, // Range for filters
                          true); // Short-cuircit for AND evaluation

                  pattern.AllowedDirection = Position_Direction;  // Set allowed
direction(s)

                  // Create signals tree
                  LogicalNode orClustersNode = new LogicalNode(Logic.Or);
                  pattern.Signals.Root.AddChild(orClustersNode);

                  // SHORT signal - cluster of buy trades
                  orClustersNode.AddChild(
                      new TradesClusterSignal(Strategy, bigTradeIndicator,
SignalDirection.Short)
                          {
                              Side = TradeSide.Ask,
                              Range = new Range() { Bars = BigTrade_ClusterBarRange, Ticks
= BigTrade_ClusterPriceRangeTicks, Logic = Logic.And },
                              MinVolume = BigTrade_MinTotalVolume,
                              MinCount = BigTrade_MinTotalCount,
                              VolumeCountLogic = Logic.And,
                              MinAggression = BigTrade_MinAggression,
                              PartiallyVisibleMode =
SignalPartiallyVisibleMode.OnSignalValidated
                          });
                  // LONG signal - cluster of sell trades
                  orClustersNode.AddChild(
                      new TradesClusterSignal(Strategy, bigTradeIndicator,
SignalDirection.Short)
                          {
                              Side = TradeSide.Bid,
                              Range = new Range() { Bars = BigTrade_ClusterBarRange, Ticks
= BigTrade_ClusterPriceRangeTicks, Logic = Logic.And },
                              MinVolume = BigTrade_MinTotalVolume,
                              MinCount = BigTrade_MinTotalCount,
                              VolumeCountLogic = Logic.And,
                              MinAggression = BigTrade_MinAggression,
                              PartiallyVisibleMode =
SignalPartiallyVisibleMode.OnSignalValidated
                          });

                  // LONG/SHORT Absoprtion SR zones signal
                  pattern.Signals.Root.AddChild(new AbsorptionSRZonesSignal(Strategy,
footPrintIndicator));

                  if (Filters_DOMImbalance)
                       pattern.Filters.Root.AddChild(new DOMImbalanceSignal(Strategy,
domImbalanceIndicator, false, Filters_DOMImbalance_Ratio));

                  Strategy.Initialize(pattern, entries, 3);  // Initialize strategy
          }
```

Note that the example above used the custom signals/filters BigTradeSignal, EMASignal, and DOMImbalanceFilter. See the next section on how to define custom signals.

CREATING CUSTOM SIGNALS

Signals are a necessary part of any advanced MZpack strategy. A signal must be placed into the decision tree as a terminal node. For example:

```
pattern.Signals.Root.AddChild(new BigTradeSignal(Strategy));
```

`BigTradeSignal` class is a custom signal declared in your code. Use `Signal` as the base class.

```
class BigTradeSignal : Signal
{
    // Constructor
    public BigTradeSignal(MZpack.NT8.Algo.Strategy strategy)
        : base(strategy,
               MarketDataSource.Level1,
               MZpack.NT8.Algo.Calculate.OnEachTick,
               false)
    {
    }
}
```

Each signal is based on one or more MZpack indicators. You can pass indicators in the constructor of the signal and store them as class members. Alternatively, you can access the indicators through the `Strategy.MZpackStrategy` property. You can find this in the samples.

Specify the attributes of your signal while calling the base constructor:

Parameter	Type	Description
strategy	MZpack.NT8.Algo.Strategy	MZpack algo strategy instance
source	MarketDataSource	Market data source for the signal: Level1 or Level2
calculate	Calculate	When the signal is calculated: OnEachTick, OnBarClose. For source = MarketDataSource.Level2 use Calculate.NotApplicable
isReset	bool	True if signal direction will be reset to None before each new calculation

Each implementation of a signal sets the properties of base `Signal` class:

Property	Type	Description
Direction	SignalDirection	Current direction of the signal: Long, Short, Any or None depending on its logic.
EntryPrice	double	Price for opening the position (**optional**)
Time	DateTime	Date and time when the signal has been validated (**optional**)
ChartRange	ChartRange	Bar/price range on the chart where the signal has been valdated (**optional**)

Implement the relevant `OnCalculate()` method to code output of the signal.

Example for `Signal.MarketDataSource == MarketDataSource.Level1`

```
public override void OnCalculate(MarketDataEventArgs e, int barIdx, SignalDirection allowed)
```

Example for `Signal.MarketDataSource == MarketDataSource.Level2`

```
public override void OnCalculate(MarketDepthEventArgs e, int barIdx, SignalDirection allowed)
```

The parameter `allowed` is the currently allowed signal/filter direction. Use it to optimize calculations.

Example of the output of a signal based on BigTrade indicator.

```
ITrade trade;  // Declare trade local variable
trade = …   // Get trade from indicator
// Set signal direction
Direction = trade.Side == TradeSide.Bid ? SignalDirection.Long : SignalDirection.Short;

// Set chart range of the signal if pattern requires localization
ChartRange = new ChartRange() { MinBarIdx = trade.StartBarIdx,
                                MaxBarIdx = trade.StopBarIdx,
                                High = trade.High,
                                Low = trade.Low };
// Set entry price of the signal
EntryPrice = trade.StopPrice;

// Add textual description for the log
Description = string.Format("{0} Big Trade signal: {1}", e.Time, trade);
```
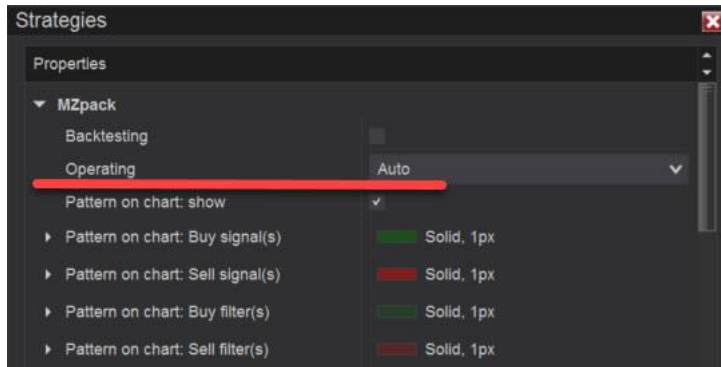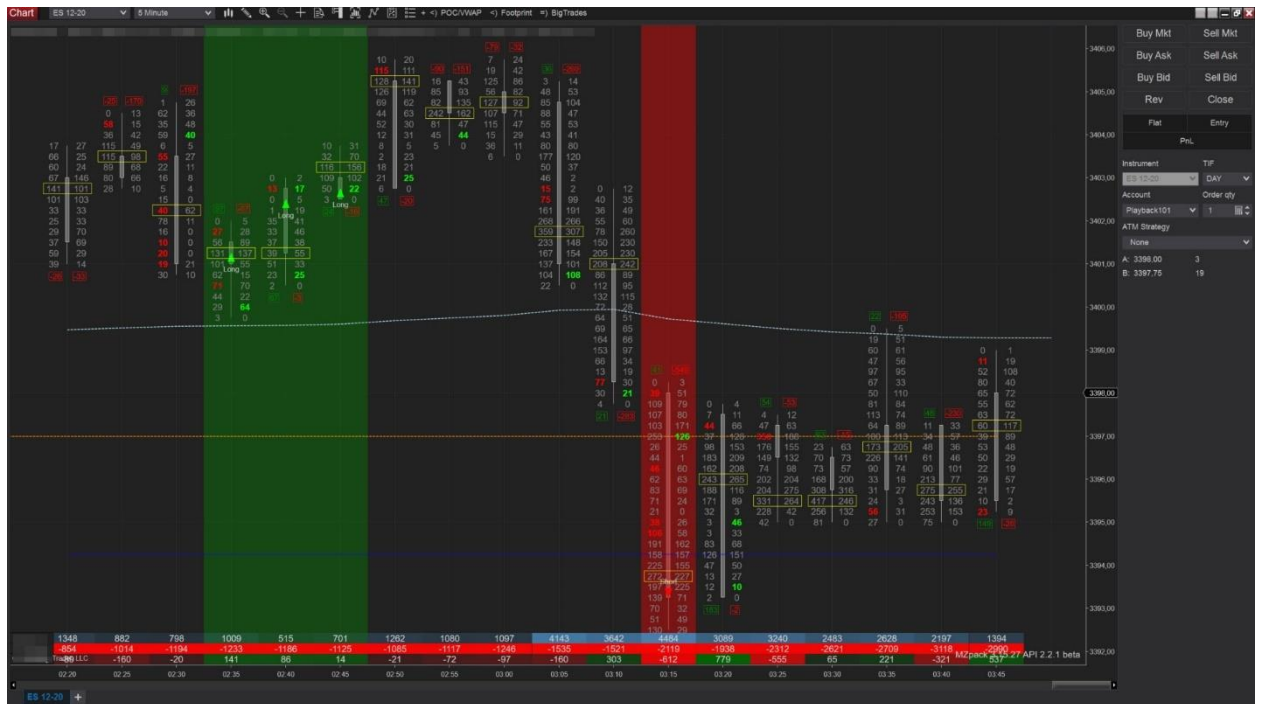
OPERATING MODES

There are two available modes for strategy behaviour: **Auto** (default) and **Manual**.

Operating mode can be set from the UI, and/or from the code using
`MZpack.NT8.Algo.MZpackStrategyBase.Operating` property.

In Manual operating mode the strategy is not making entries, so you can trade manually by the signals. See an example of visualization of patterns.

VISUALIZATION OF PATTERNS

There is a **Visual** category in the strategy properties created by the MZpack API. To display vertical area under validated Entry patterns on the chart, enable the **Pattern: background** option. The area under validated pattern will have the selected **colors** for **Long** and **Short** directions. As you know, a pattern may consists of two parts: signals and filters. You can assign different background colors for each of them.

When the strategy is in **Manual** operating mode, entry and exit markup can be useful. Set **Entry/Exit: markup** = **Marker** or **MarkerAndText**.
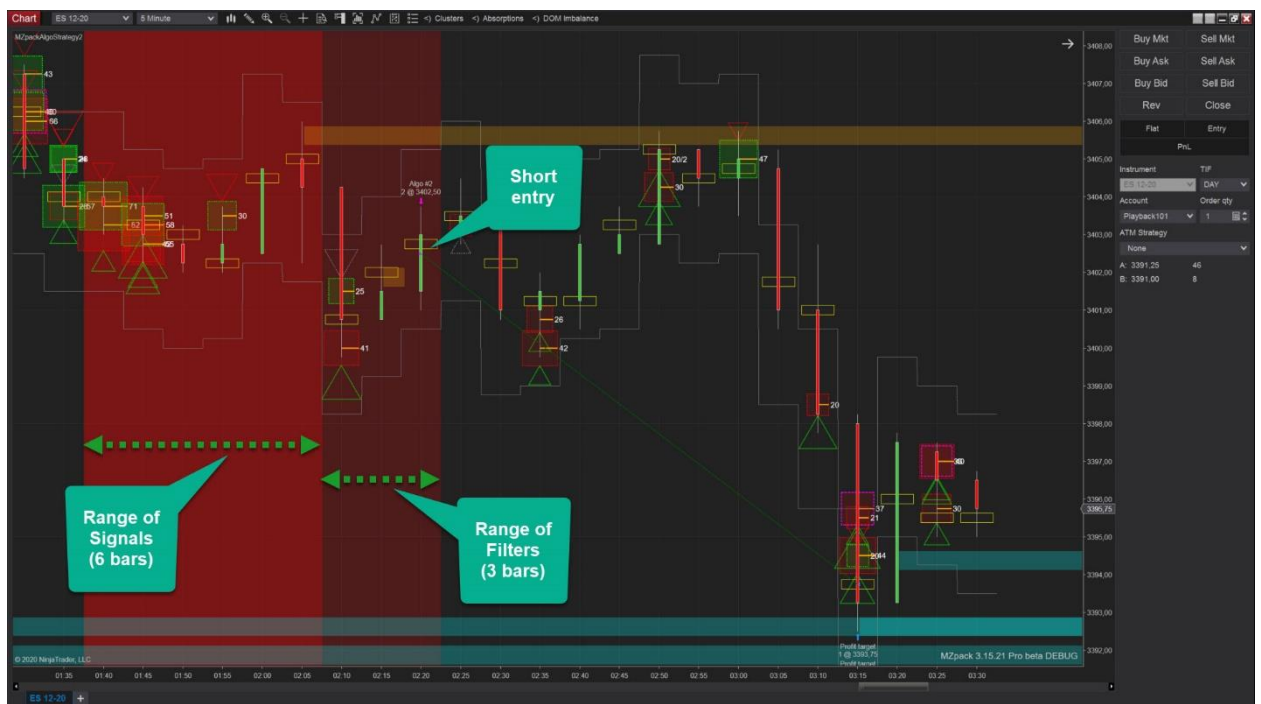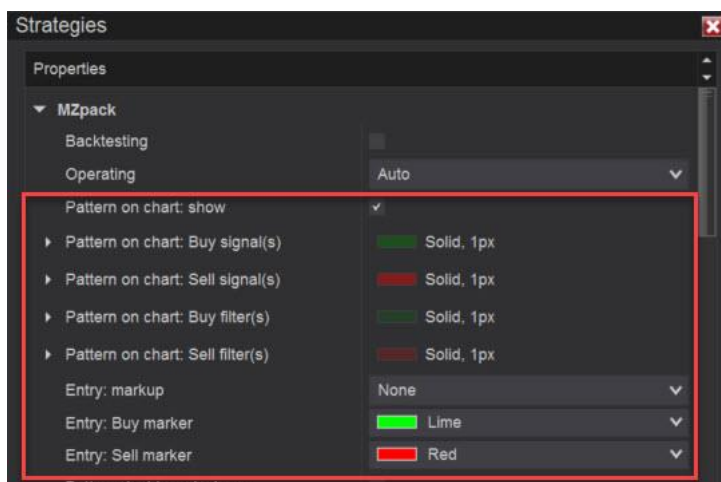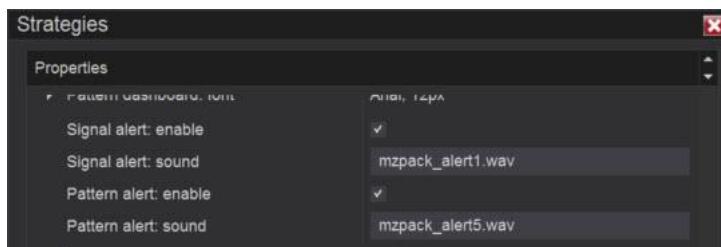
I this particular example we used clusters of big trades and absorption zones as signals, and the DOM imbalance as a filter (see **MZpackAlgoStrategy2.cs** sample). If you run this strategy, the resulting image will appear quite confusing because the plots of all the indicators will be visible on the chart.

Enable **Pattern dashboard: show** option to see logical tree of Entry pattern on the chart along with signals/filters states on each bar. You can use it in Manual operating mode for your discretionary trading.
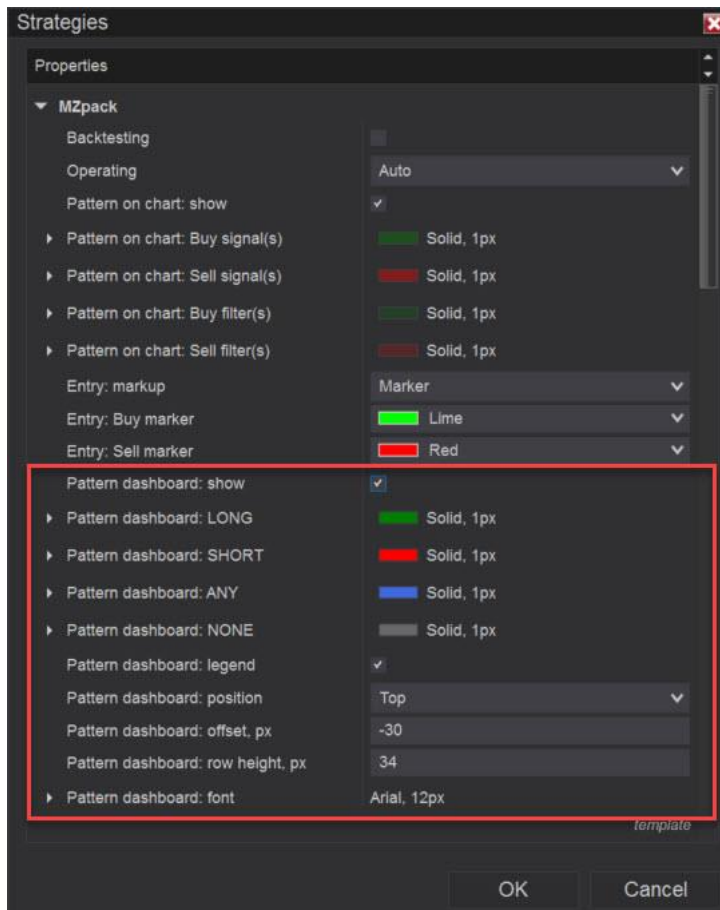
Alerts can be set upon validating signal/filter and whole pattern. Sound files can be one of pre-installed .wav files: mazpack_alert1.wav, mazpack_alert2.wav,.. mazpack_alert8.wav. Or assign your own .wav file.

You can find alerts in Alert Log window.

To get a clearer picture, use the **Partially Visible** mode for the indicators. Click on the "eye" button next to the indicator name to switch to partially visible mode.
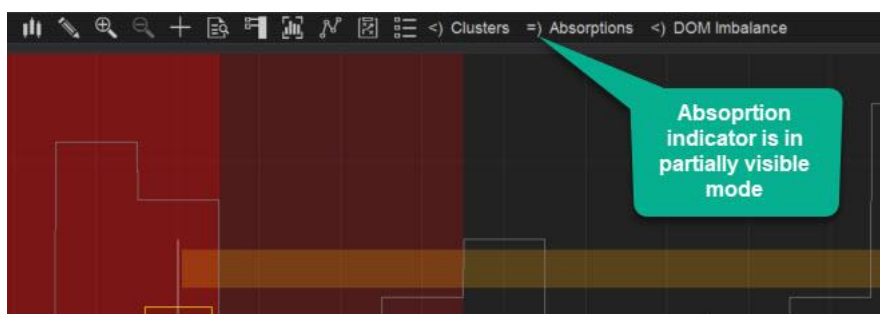
The Partially Visible mode is only available for API versions of the following indicators.

Indicator	Interface and property
StartegyBigTradeIndicator	ITrade.View.PartiallyVisible
StartegyFootprintIndicator	IFootprintBar.PartiallyVisible
StartegyMarketDepthIndicator	IMarketDepthBlock.PartiallyVisible

See MZpackAlgoStrategy2.cs sample on how to code this feature. There are two methods to show plots.

1. Show plots of the signal during pattern validation process.

```csharp
// Declare our custom signal
class AbsorptionSRZonesSignal : Signal
{
    IFootprintBar bar;  // Private member to store validation bar

    public override void OnCalculate(MarketDataEventArgs e, int barIdx,
SignalDirection allowed)
    {
        // Here is the logic of the signal, assign bar member here
        // …
        // Make footprint bar partially visible when signal is validated
        bar.PartiallyVisible = true;
    }
```

2. Show plots of the signal on validation of the whole pattern.

```csharp
    public override void OnPatternValidated()
    {
        if (bar != null)
            bar.PartiallyVisible = true; // Show footprint bar in partially
visible mode
    }
```

For both methods, add code to hide your plots inside the Reset() overrider.

```csharp
    protected override void Reset(bool resetView)
    {
        base.Reset(resetView);  // base call required

        if (bar != null)
```

```
                {
                    // Make footprint bar not visible in partially visible mode
                    // on signal reset
                    if (resetView)  // Check resetView state
                        bar.PartiallyVisible = false;
                    bar = null;
                }
            }
```

Our chart now looks cleaner.

You can change modes of visibility back and forth from within the UI.

States of visibility, however, require additional code in order to be serialized in a chart or workspace template. Declare public properties for states, and place those, for example, in the **Visual** category.

```
        [Display(Name = "Absorptions: visible", GroupName = "Visual", Order = 1002,
Description = "")]
        public bool VisibleFootPrintIndicator { get; set; }
        [Display(Name = "Absorptions: partially visible", GroupName = "Visual", Order =
1003, Description = "")]
        public bool PartiallyVisibleFootPrintIndicator { get; set; }
```

In `OnStateChange()` add the code to restore states on `State == State.Configure`, and to save states on `State == State.Terminated`.

```
            if (State == State.Configure)
            {
                footPrintIndicator.Visible = VisibleFootPrintIndicator;
                footPrintIndicator.IsPartiallyVisible =
PartiallyVisibleFootPrintIndicator;
```

```
            }
            else
            if (State == State.Terminated)
            {
                if (footPrintIndicator != null)
                {
                    VisibleFootPrintIndicator = footPrintIndicator.Visible;
                    PartiallyVisibleFootPrintIndicator =
footPrintIndicator.IsPartiallyVisible;
                }
            }
```

You can control visualization of some of built-in signals through `SignalPartiallyVisibleMode` property of `Signal` class. See `TradesClusterSignal` class as an example.

TRADING TIMES

The strategy doesn't make entries outside Trading Times. Working orders outside Trading Times will be closed, pending orders will be canceled. Trading Times doesn't affect on a strategy in Manual operating mode.

See TradingTimes sample strategy for the details.

Auto operating mode. Trading Times set 08:30 – 11:00

Manual operating mode. Trading Times set 08:30 – 11:00

RISK MANAGEMENT

Risk management is implemented by RiskManagement class. It allows to set daily profit/loss and trades limits. The daily risk is reset overnight if the strategy is not in position.

The typical code is

```
protected MZpack.NT8.Algo.Strategy CreateAlgoStrategy()
        {
            MZpack.NT8.Algo.Strategy strategy = new MZpack.NT8.Algo.Strategy(@"My
Strategy", this)
            {
                OppositePatternAction = OppositePatternAction.None,
                LogLevel = LogLevel, // Set log level of algo from strategy UI
                LogTarget = LogTarget, // Set log target of algo from strategy UI
                LogTime = LogTime
            };

            // Risk
            strategy.RiskManagement = new MZpack.NT8.Algo.RiskManagement(strategy)
            {
                Currency = Currency.UsDollar,  // Currency of the account
                EntryName = ENTRY_NAME,  // Name of the entry to monitor trades number.
                DailyLossLimitEnable = Common_DailyLossLimit_Enable,
                DailyLossLimit = Common_DailyLossLimit,
                DailyProfitLimitEnable = Common_DailyProfitLimit_Enable,
                DailyProfitLimit = Common_DailyProfitLimit,
                DailyTradesLimitEnable = Common_DailyTradesLimit_Enable,
                DailyTradesLimit = Common_DailyTradesLimit
            };

            return strategy;

        }
```

See RiskManagement for the details.

MAKE YOUR STRATEGY READY FOR BACKTESTING

Some MZpack indicators have load time optimization. This means that the final calculation of some indicator values occurs at the end of loading the historical data. This is the default behavior. Also, MZpack strategy is not active in historical state by default. However, in the Strategy Analyzer, backtesting works in historical state only.

Enable option **MZpack: backtesting** in the strategy settings window.
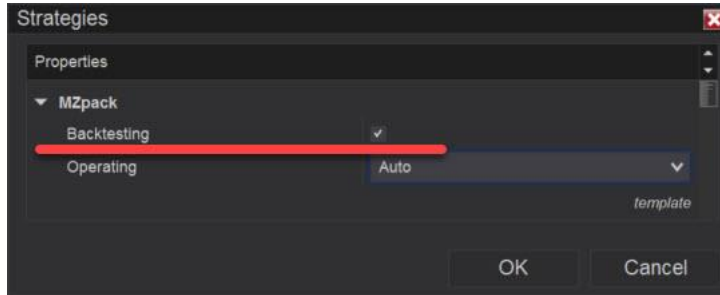
If you want to enable support of backtesting by default, from the code, set the property `MZpackStrategyBase.EnableBacktesting = true` on `State == State.SetDefaults`.

```
protected override void OnStateChange()
            if (State == State.SetDefaults)
            {
                // This is required for backtesting in Strategy Analyzer
                EnableBacktesting = true;
            }
```

This will set the property of used indicators `ModelIncrementRefresh = ModelIncrementRefresh.HistoricalRealtime`. Also, historical working state for the strategy will be set `WorkingStateHistorical = true.` You don't need to set these values manually from the code.

Support of the backtesting will increase loading time for these indicators if some of the features are enabled: mzFootprint, mzVolumeProfile.

BUILT-IN SIGNALS AND FILTERS

Use built-in signals from `MZpack.NT8.Algo.Signals` namespace as building blocks for your strategies.

| Signal class	Base indicator(s)	Description	Entry price
TradesClusterSignal	mzBigTrade	Detects cluster of trades (group of trades) at the given side within given bar/price ranges	POC of the last trade in the cluster
DOMImbalanceSignal	mzMarketDepth	Detects DOM imbalance	Optional. Best bid for LONG, Best offer for SHORT
DOMBlockSignal	mzMarketDepth	Detects large DOM blocks. **Allowed direction can't be ANY**	Optional, used block price that is nearest to the best price. LONG for bid blocks, and SHORT for offer blocks.
FootprintImbalanceSignal	mzFootprint	SHORT for Sell imbalance/LONG for Buy imbalance	Current price or bar close
FootprintSRZonesSignal	mzFootprint	Searches for S/R Zones of the given type: Imbalance or Absorption. Generates Long direction for BUY Imbalance S/R Zone(s) and Short direction for SELL Imbalance S/R zone(s). Generates Long direction for SELL Absoprtion S/R Zone(s) and Short direction for BUY Absoprtion S/R zone(s).	Current price or bar close

Note, to get entry price from the signal, set HasPrice = true for it. Override `GetEntryPrice(object args)` method (where it is used) to implement your own calculation for the entry price.

You can find the source code for all bnuilt-in signals inside the MZpack bundle 3 (Pro or Standard) installation folder, inside **\API\signals** folder.

LOGGING

All advanced MZpack strategies have built-in logging options. Logging options are available from UI by default.
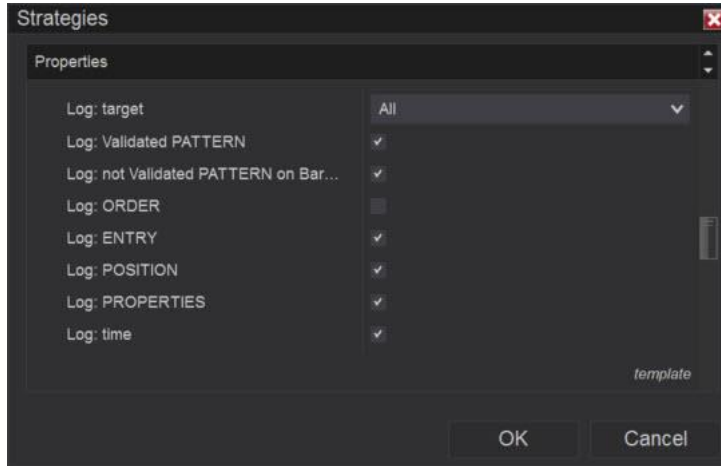
Specify the logging parameters while creating an instance of `MZpack.NT8.Algo.Strategy` class if you need specific logging behaviour.

```
protected MZpack.NT8.Algo.Strategy CreateAlgoStrategy()
{
    return new MZpack.NT8.Algo.Strategy(@"My MZpack Advanced Strategy",
        this)
    {
        LogLevel = LogLevel.ALL, // Log all events
        LogTarget = LogTarget.NinjaScriptOutput, // Log to NinjaScript output
                                                 Only
        LogTime = true  // Log chart date and time
    };
}
```

LogLevel value	Level
NONE	No logging
NV_PATTERN_OBC	Log not validated pattern on each bar close. Use it to debug your strategy
V_PATTERN	Log pattern when validated
ORDER	Submitted, Working, Filled, PartFilled, Cancelled, Rejected orders
ENTRY	Entries of the strategy
POSITION	Position state
PROPERTIES	Properties with NinjaScriptProperty attribute
ALL	Log all events above

For example, to log all events except not validated patterns on bar close, use this code

```
LogLevel = LogLevel.ALL & ~LogLevel.NV_PATTERN_OBC;
```

LogTarget value	Target
LogTarget.NinjaScriptOutput	NinjaScript Output Window
LogTarget.File	Documents\NinjaTrader 8\mzpacklog\<file>.txt
LogTarget.All	NinjaScript Output Window + File

Here is the log file name example: `strategy.My MZpack Algo Strategy.12122020.020723.txt`, where `My MZpack Algo Strategy` is the name of the strategy; 12122020 is the current systm date.

Log example:

```
27.10.2020 8:29:02 NV_PATTERN_OBC, V_PATTERN:: [ Long] [VALIDATED]
27.10.2020 8:29:02 NV_PATTERN_OBC, V_PATTERN:: SIGNALS: Chart range: Bars 370-373 @3399-
3400,5
27.10.2020 8:29:02 NV_PATTERN_OBC, V_PATTERN:: Logic [And, Long]
27.10.2020 8:29:02 NV_PATTERN_OBC, V_PATTERN::   Logic [Or, Long]
27.10.2020 8:29:02 NV_PATTERN_OBC, V_PATTERN::     Signal [TradesClusterSignal, None]
27.10.2020 8:29:02 NV_PATTERN_OBC, V_PATTERN::     Signal [TradesClusterSignal, Long]
Chart range: Bars 370-370 @3400,5-3400,5 Clusters: 3 Bid trades, total 112-Lot @3400,5-
3400,5 trades: 20+35+57
27.10.2020 8:29:02 NV_PATTERN_OBC, V_PATTERN::   Signal [AbsorptionSRZonesSignal, Long]
Chart range: Bars 373-373 @3399-3399 AbsorptionSRZonesSignal: Buy absoprtion zone(s) 0,
Sell absoprtion zone(s) 1
27.10.2020 8:29:02 NV_PATTERN_OBC:: FILTERS: Chart range: Bars 373-374 @0-0
27.10.2020 8:29:02 NV_PATTERN_OBC, V_PATTERN:: Logic [And, Long]
27.10.2020 8:29:02 NV_PATTERN_OBC, V_PATTERN::   Signal [DOMImbalanceSignal, Long] Chart
range: Bars 374-374 @0-0 DOM Imbalance: 1,46539027982327
27.10.2020 8:29:02 POSITION:: Position state: EntrySubmitting
27.10.2020 8:29:02 ENTRY:: 27.10.2020 8:29:02 'Algo #2' LONG Market @3399 2-Lot
27.10.2020 8:29:03 ORDER:: Order submitted 'Algo #2'
orderId='d0fa442e90874deab7363fd3abba2bdf' account='Playback101' name='Algo #2'
orderState=Submitted instrument='ES 12-20' orderAction=Buy orderType='Market'
limitPrice=0 stopPrice=0 quantity=2 tif=Gtc oco='' filled=0 averageFillPrice=0
onBehalfOf='' id=888 time='2020-10-27 08:29:03' gtd='2099-12-01' statementDate='2020-10-
27'
27.10.2020 8:29:03 POSITION:: Position state: LongMarketPending
27.10.2020 8:29:03 ORDER:: Order placed 'Algo #2'
orderId='d0fa442e90874deab7363fd3abba2bdf' account='Playback101' name='Algo #2'
orderState=Working instrument='ES 12-20' orderAction=Buy orderType='Market' limitPrice=0
stopPrice=0 quantity=2 tif=Gtc oco='' filled=0 averageFillPrice=0 onBehalfOf='' id=888
time='2020-10-27 08:29:03' gtd='2099-12-01' statementDate='2020-10-27'
27.10.2020 8:29:03 ORDER:: Order filled 'Algo #2'
orderId='d0fa442e90874deab7363fd3abba2bdf' account='Playback101' name='Algo #2'
orderState=Filled instrument='ES 12-20' orderAction=Buy orderType='Market' limitPrice=0
stopPrice=0 quantity=2 tif=Gtc oco='' filled=2 averageFillPrice=3398.75 onBehalfOf=''
id=888 time='2020-10-27 08:29:03' gtd='2099-12-01' statementDate='2020-10-27'
POSITION:: Position state: Long
27.10.2020 8:31:51 ORDER:: Exit entry 'Algo #2'
orderId='2976bdea8bed4700893ee7f4223515f5' account='Playback101' name='Stop loss'
orderState=Filled instrument='ES 12-20' orderAction=Sell orderType='Stop Market'
```

```
limitPrice=0 stopPrice=3394.25 quantity=2 tif=Gtc oco='c1c1f73af2f740dda070d894e6f1ff06'
filled=2 averageFillPrice=3394.25 onBehalfOf='' id=889 time='2020-10-27 08:31:51'
gtd='2099-12-01' statementDate='2020-10-27'
```

```
POSITION:: Position state: Flat
```

THE SAMPLES

There are two types of MZpack API samples: data access and strategies. The samples are all precompiled so you can run it in NinjaTrader. You can modify and compile the samples using source code.

The source code for API samples can be found inside the MZpack 3 (Pro or standard) bundle installation folder, inside the **\API\samples** folder. Also, you can download the samples files in zip-archive from the site. DON'T IMPORT THIS FILE INTO NinjaTrader. Unzip it to a folder and learn the API usage.

Version of samples on the site may be not actual. We recommend reviewing the samples from the installation folder.

How to Modify and Compile a Sample

1. Copy sample .cs file (eg. MZpackAlgoStrategy0.cs) into the **Documents\NinjaTrader 8\bin\Custom\strategies** folder.
2. Open NinjaScript editor. Select the copied strategy from the list of Strategies.
3. Remove **#if APISAMPLE** and **#endif** directives from the source code.
4. Add some name to strategy namespace, for example, JohnDow:
   ```
   namespace NinjaTrader.NinjaScript.Strategies.JohnDow
   ```
5. Add a reference to MZpack.NT8.Pro.dll in your project. In NinjaScript editor, choose *References...* from the context menu. Click *add* button and choose MZpack.NT8.Pro.dll file, then click Open. Click OK to close References window.
6. Compile the strategy. You will see your strategy under JohnDow folder in the list of available strategis.
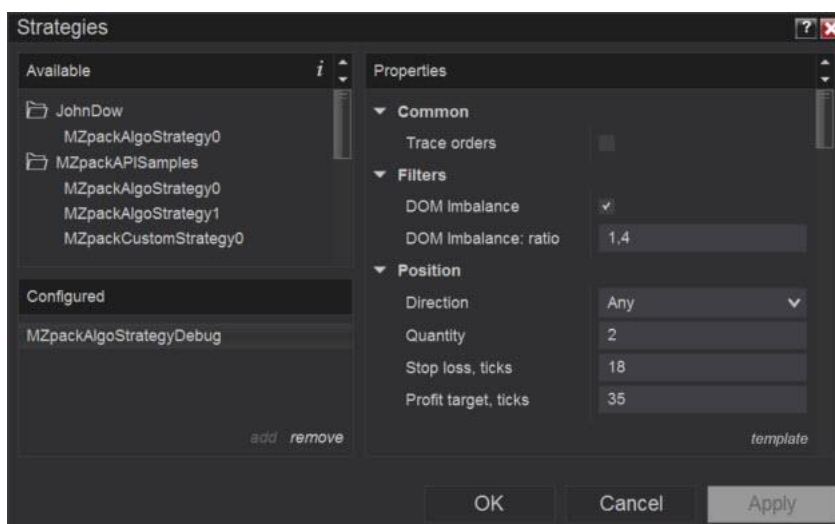
Demo Strategies

SIMPLE E-MINI S&P500 FUTURES SCALPING STRATEGY (NOT AUTOMATED)

This simple scalping strategy uses information regarding Market Microstrucuture and Dynamic volumes based day levles (dynamic volume POC, VAL/VAH, VWAP and deviations).

Read a description of the strategy at: https://www.mzpack.pro/2018/02/17/simple-e-mini-sp500-futures-scalpingstrategy/