



NINJATRADER 8 INDICATOR · NINJAScript, C#

User Guide

Pivot trendlines, breakout signals scored by four confluence filters, an ATR stop with three targets, a live dashboard, alerts, and series a strategy can read.

PRODUCT	Apex Break PRO for NinjaTrader 8
VERSION	1.0.1
DATE	18 September 2026
NINJATRADER	8.1.x Desktop, tested on 8.1.8.2
PACKAGE	ApexBreakPro.zip , compiled NinjaScript Add-On
SHA - 256	f1d6fc90ceb618f85304d16ff9bd0e67a46d402c98d2e00de266970f277fcba7
AUTHOR	Kestutis Balciunas, https://kestutisbalciunas.com/
PRODUCT PAGE	https://kestutisbalciunas.com/ninjatrader/apex-break-pro/
SUPPORT	https://kestutisbalciunas.com/contact/

Trading involves substantial risk. Futures, foreign currency and options trading contains substantial risk and is not for every investor. Past, simulated and backtested performance is not necessarily indicative of future results.

Contents

01 What it is

02 Installation

03 Quick start

04 Calculate mode and bar timing

05 Outputs for strategies and the Market
Analyzer

06 Parameters

07 Sessions, time and volume

08 Compatibility

09 Troubleshooting

10 How the port was verified

11 Licensing

12 Support

13 Risk disclosure

What it is

Chart indicator for NinjaTrader 8 Desktop (NinjaScript, C#): pivot trendlines drawn through consecutive confirmed pivot highs and pivot lows, breakout signals when a closed bar breaks an active line by more than an ATR margin, scored by four confluence filters (EMA trend, volume spike, RSI boundary, ADX strength) into a five-star rating, an ATR (or percent) stop with three risk-reward targets, a two-column dashboard and nine alert conditions. Signals only: it never places, modifies or closes orders.

Installation

Requirements

- NinjaTrader 8.1.x Desktop for Windows (tested on 8.1.8.2). Not for NinjaTrader 8.0 or NinjaTrader Web.
- Any data connection that delivers bars for the chart (the indicator needs no Order Flow+ features, no depth, no tick history). Volume is used when the feed supplies it.

Import the Add-On package

1. Control Center > Tools > Import > NinjaScript Add-On.
2. Select `ApexBreakPro.zip` and confirm. NinjaTrader compiles the assembly and reports "Import complete".
3. No restart is needed. If a chart already had an older version, remove and re-add the indicator (a compile does not reload running instances).

Verify installation

1. Open a chart (any instrument, a 5 to 60 minute interval is a good first look), press Ctrl+I.
2. In the Available list find **Apex Break PRO**, click add, click OK.
3. The chart shows the pivot trendlines (solid while active, dashed once broken or expired), the colour-coded trend EMA, the BUY / SELL diamonds with their three-line labels, the entry / SL / TP lines of the last signal with the risk and reward zones, and the dashboard in the top-right corner. With 200 bars or fewer loaded the EMA and the first trendlines take a while to appear: pivots need `Pivot Left Bars + Pivot Right Bars` bars and the EMA its length, so load a few days of history.

Uninstall / update

- Remove: Control Center > Tools > Remove NinjaScript Assembly > Apex Break PRO.
- Update: import the new zip over the installed version (NinjaTrader replaces the assembly), then remove and re-add the indicator on every chart that uses it, or reload the workspace.

Quick start

First chart

1. Add **Apex Break PRO** to a chart (Ctrl+I). Leave every setting at its default.
2. Load enough history: with the defaults the EMA needs 200 bars and every trendline needs a pair of pivots, each confirmed 20 bars after the fact. A few days of 5-minute bars is plenty.
3. Signals print on closed bars only. Wait for a close that breaks an active trendline.

Reading the chart

- **Pivot trendlines:** every new confirmed pivot high draws a resistance line through the previous pivot high, every pivot low a support line. Solid while active; dashed once price closed through it (a breakout) or once it is older than **Max Line Extension**. Broken and expired lines stay on the chart until the oldest are dropped by **Max Resistance / Support Lines**.
- **Trend EMA:** one line, green while the close is above it, red while below (the EMA Trend Filter).
- **Diamond and label:** a close beyond an active resistance line (plus the breakout margin) that passes the enabled filters and the cooldown is a BUY, with a diamond under the bar and a three-line label: the banner, the close, and the star rating with the confluence fraction, for example ★★☆☆ (3/4) . A SELL is the mirror on a support line. Signals never repaint.
- **Entry / SL / TP1 / TP2 / TP3:** drawn on every signal while **Show TP/SL Levels** is on. The stop is $SL \text{ ATR Multiplier} \times ATR$ from the entry (or a percentage), the targets sit at the three risk-reward ratios. The lines extend to the right of the current bar while the trade is tracked, the red zone is the risk, the green zone the reward to the furthest shown target, and the price tags move with the current bar. When the stop or the third target is reached the drawing stops at that bar.
- A hit target is not restyled on the chart; the check mark appears in the dashboard TRADE rows, exactly as in the original.

Signal rules in one paragraph

A raw break is a close above an active resistance line (or below a support line) by more than $Margin \times ATR$ when **Min Breakout Margin** is on. Each enabled filter adds one point of confluence: close above the EMA (BUY) or below (SELL); volume above $Volume \text{ Multiplier} \times$ its average; RSI below the overbought level (BUY) or above the oversold level (SELL); ADX above its threshold. A break becomes a signal when the score reaches **Min Confluence Score** and no signal of the same side printed within **Signal Cooldown** bars. The stars are $round(score \times 5 / \text{enabled filters})$.

The dashboard

Two columns in the corner chosen by **07 Dashboard > Position** (six positions). It always reflects the most recent bar. Rows for a disabled filter are absent, so the table is 15 rows with the default three filters and 16 with the ADX filter enabled.

ROW	COLUMN 1	COLUMN 2
1	Kestutis Balciunas	Apex Break PRO v1.0.0
2	SIGNAL	section header
3	Direction	▲ LONG (Active), ▼ SHORT (Closed) or – WAITING
4	Strength	★★★☆☆ (3/4) : the star rating and the confluence fraction of the last signal
5	FILTERS	section header
6	EMA (200)	✓ Above 29390.00 or X Below ... : which side of the EMA price is on, and the EMA value
7	RSI (14)	54.3 ✓ : the RSI value, ticked when it sits between the oversold and overbought levels
8	Volume	✓ Spike 1.4x or X Low 0.9x : this bar's volume as a multiple of its average
9	ADX (14)	22.3 ✓ , only when the ADX filter is enabled
10	TRADE	section header
11	Entry	the close of the signal bar
12	SL	the stop
13	TP1	29362.25 (1:1) , with ✓ once price has reached it
14	TP2	(1:2) , likewise
15	TP3	(1:3) , likewise
16	Kestutis Balciunas	v1.0.0

The FILTERS rows are live and update every bar; the SIGNAL and TRADE rows are frozen at the last signal, so the two can disagree, which is how you see conditions changing since the signal. A dash in the TRADE rows means no trade has ever been tracked on this chart; if signals print and the rows stay dashed, **Show TP/SL Levels** is off, and in that state the original script tracks no trade at all.

Alerts

Nine events, each behind its own toggle in **08 Alerts**, evaluated once per closed bar and only for bars that close after the indicator went live (never during the history pass). They land in the Alerts Log with an optional sound (Sound File inside NinjaTrader's sounds folder) and an optional email through NinjaTrader's Mail share service.

TOGGLE	DEFAULT	FIRES WHEN
Alert: Buy Signal	on	a confirmed close breaks a resistance line and passes the confluence and cooldown tests
Alert: Sell Signal	on	the same on the support side
Alert: Any Signal	off	either of the two above
Alert: Resistance Break	off	a resistance line is broken, before the filters are applied
Alert: Support Break	off	a support line is broken, before the filters
Alert: TP1 Hit	on	price reaches the first target for the first time
Alert: TP2 Hit	on	the second target

TOGGLE	DEFAULT	FIRES WHEN
Alert: TP3 Hit	on	the third target, which also closes the tracked trade
Alert: Stop Loss Hit	on	price reaches the stop, which also closes the tracked trade

The buy and sell bodies are the original's text (BUY | MNQ | TF: 5 | Price: ... | ★★☆☆ Moderate (3/4) | SL: ... | TP1: ... | TP2: ... | TP3: ...) or, with **Webhook JSON Format** on, the same values as JSON. Nothing is sent anywhere by the indicator itself.

Using the values in a strategy or the Market Analyzer

Three series are exposed: `TrendEma` (the EMA, NaN while hidden or unseeded), `Signal` (+1 on a BUY bar, -1 on a SELL bar, 0 otherwise) and `TradeDirection` (+1 while a long trade is tracked, -1 for a short, 0 when flat). Example: `if (ApexBreakPro(...).Signal[0] > 0) .`

04

Calculate mode and bar timing

- Default `Calculate.OnBarClose`. Breakouts and signals are evaluated on closed bars only (Pine `barstate.isconfirmed`).
- `OnEachTick` / `OnPriceChange`: the forming bar is previewed on every tick on a scratch copy of the committed state, exactly as TradingView rolls the script back before each realtime tick: pivots, line creation, line expiry, the confluence rows of the dashboard and TP/SL hit detection all run on the forming bar, as they do in the original; the break test and the alerts wait for the close. The bar is committed once at `IsFirstTickOfBar` of the next bar. Signals never repaint; trendlines are provisional on the forming bar on both platforms.
- Historical pass: one `OnBarUpdate` per closed bar. The last historical bar (still forming when the chart loads) is previewed, not committed, and committed by the first realtime close. The engine slot is derived from `CurrentBar`, so the historical to realtime transition can neither double-feed nor skip a bar; a skipped range is refilled from the `Bars` series.
- Tick Replay: handled as per-tick realtime.
- The indicator keeps calculating while its chart window is minimized (`IsSuspendedWhileInactive = false`), so alerts fire on time; and an alert is only raised for a bar that closed within the last two bar periods, so a catch-up after a suspension or a reconnect never replays old alerts.

05

Outputs for strategies and the Market Analyzer

PLOT / SERIES	VALUES
<code>TrendEma</code> (plot 0)	the trend EMA, coloured by the close's side per bar; NaN while unseeded or hidden

PLOT / SERIES	VALUES
Signal (plot 1, hidden)	+1 on a confirmed BUY bar, -1 on a confirmed SELL bar, 0 otherwise
TradeDirection (plot 2, hidden)	+1 while a long trade is tracked, -1 for a short, 0 when flat

Example in a strategy: `var apx = ApexBreakPro(...); if (apx.Signal[0] > 0) ...` (the accessor is generated by NinjaTrader on compile).

Parameters

The 46 inputs of the original script keep their titles, groups, order and defaults (the group names carry a number prefix because NinjaTrader sorts groups alphabetically). The 19 NinjaTrader-only additions (the original exposes its nine alert conditions through TradingView's alert dialog) are the nine per-event alert toggles and the sound and email settings in group 08 Alerts, and the marker style, label cap, trendline cap, draw-ahead distance and the two diagnostic switches in group 10 NinjaTrader: 65 parameters in total, the same set as the cTrader edition.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
01 Pivot Detection	Pivot Left Bars	int	20	5 to 50	PivotLeftBars	
01 Pivot Detection	Pivot Right Bars	int	20	5 to 50	PivotRightBars	
01 Pivot Detection	Max Line Extension (bars)	int	500	10 to 500	MaxLineExtension	
02 Line Display	Max Resistance Lines (0 = all)	int	5	0 to 20	MaxResistanceLines	0 means all in the original; here 0 is capped by Max Trend Lines Per Side.
02 Line Display	Max Support Lines (0 = all)	int	5	0 to 20	MaxSupportLines	0 means all in the original; here 0 is capped by Max Trend Lines Per Side.
02 Line Display	Line Width	int	3	1 to 5	LineWidth	
02 Line Display	Line Transparency (0-100)	int	25	0 to 100	LineTransparency	
03 Signal Filters	EMA Trend Filter	bool	true		UseEmaFilter	
03 Signal Filters	EMA Length	int	200	10 to 500	EmaLength	
03 Signal Filters	Show EMA on Chart	bool	true		ShowEmaOnChart	
03 Signal Filters	Volume Spike Filter	bool	true		UseVolumeFilter	
03 Signal Filters	Volume Multiplier	double	1.2	0.5 to 3.0	VolumeMultiplier	
03 Signal Filters	Volume Average Length	int	20	5 to 100	VolumeAvgLength	
03 Signal Filters	RSI Boundary Filter	bool	true		UseRsiFilter	
03 Signal Filters	RSI Length	int	14	5 to 50	RsiLength	

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
03 Signal Filters	Overbought Level	int	75	60 to 90	RsiOverbought	
03 Signal Filters	Oversold Level	int	25	10 to 40	RsiOversold	
03 Signal Filters	ADX Trend Strength Filter	bool	false		UseAdxFilter	
03 Signal Filters	ADX Length	int	14	5 to 50	AdxLength	
03 Signal Filters	ADX Threshold	double	20.0	10.0 to 50.0	AdxThreshold	
03 Signal Filters	Min Breakout Margin	bool	true		UseBreakoutMargin	
03 Signal Filters	Margin (x ATR)	double	0.1	0.01 to 1.0	MarginMultiplier	
03 Signal Filters	Signal Cooldown (bars)	int	5	0 to 50	SignalCooldown	
03 Signal Filters	Min Confluence Score	int	2	0 to 4	MinConfluence	
04 Signal Display	Show Signal Arrows	bool	true		ShowArrows	
04 Signal Display	Show Signal Labels	bool	true		ShowLabels	
05 Risk Management	Show TP/SL Levels	bool	true		ShowTpSl	Gates the whole trade engine, as in the original: with it off no trade is tracked, the dashboard TRADE rows stay empty and the alert bodies print NaN for SL and TP.
05 Risk Management	SL = ATR x Multiplier	bool	true		UseAtrSl	
05 Risk Management	ATR Period	int	14	1 to 500	AtrPeriod	
05 Risk Management	SL ATR Multiplier	double	1.5	0.1 to 20.0	SlMultiplier	
05 Risk Management	SL % (if not ATR)	double	1.0	0.05 to 50.0	SlPercent	
05 Risk Management	TP1 Risk-Reward	double	1.0	0.1 to 50.0	Tp1Rr	
05 Risk Management	TP2 Risk-Reward	double	2.0	0.1 to 50.0	Tp2Rr	
05 Risk Management	TP3 Risk-Reward	double	3.0	0.1 to 50.0	Tp3Rr	
06 TP/SL Display	Show SL Level	bool	true		ShowSl	
06 TP/SL Display	Show TP1 Level	bool	true		ShowTp1	
06 TP/SL Display	Show TP2 Level	bool	true		ShowTp2	
06 TP/SL Display	Show TP3 Level	bool	true		ShowTp3	
07 Dashboard	Show Dashboard	bool	true		ShowDashboard	
07 Dashboard	Position	ApxDashPos	TopRight		DashboardPosition	

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
08 Alerts	Enable Alerts	bool	true		EnableAlerts	
08 Alerts	Webhook JSON Format	bool	false		WebhookJson	Signal alert body as JSON instead of text (the message only; no network call is made).
08 Alerts	Alert: Buy Signal	bool	true		AlertBuy	
08 Alerts	Alert: Sell Signal	bool	true		AlertSell	
08 Alerts	Alert: Any Signal	bool	false		AlertAny	
08 Alerts	Alert: Resistance Break	bool	false		AlertResBreak	A resistance line is broken on a closed bar, before the filters are applied.
08 Alerts	Alert: Support Break	bool	false		AlertSupBreak	A support line is broken on a closed bar, before the filters are applied.
08 Alerts	Alert: TP1 Hit	bool	true		AlertTp1	
08 Alerts	Alert: TP2 Hit	bool	true		AlertTp2	
08 Alerts	Alert: TP3 Hit	bool	true		AlertTp3	
08 Alerts	Alert: Stop Loss Hit	bool	true		AlertSl	
08 Alerts	Play Sound	bool	true		PlaySound	
08 Alerts	Sound File	string	"Alert2.wav"		AlertSoundFile	File name inside the NinjaTrader sounds folder, for example Alert2.wav.
08 Alerts	Send Email	bool	false		SendEmail	Uses the Mail share service configured in NinjaTrader (Tools > Options > Share).
08 Alerts	Email To	string	""		EmailTo	
09 Colors	Resistance Line Color	colour	#EF5350		ResistanceBrush	
09 Colors	Support Line Color	colour	#26A69A		SupportBrush	
09 Colors	Buy Signal Color	colour	#00E676		BuySignalBrush	
09 Colors	Sell Signal Color	colour	#FF1744		SellSignalBrush	
10 NinjaTrader	Signal Marker Style	ApxMarkerStyle	DiamondAndPill		MarkerStyle	The original draws a diamond carrying the text plus a separate three-line label; this selects which of the two appear.
10 NinjaTrader	Max Signal Labels	int	300	50 to 500	MaxSignalLabels	Oldest signal markers are removed beyond this count (the original keeps 500).

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
10 NinjaTrader	Max Trend Lines Per Side	int	100	10 to 500	MaxTrendLinesPerSide	The cap applied when Max Resistance/Support Lines is 0 (all).
10 NinjaTrader	Draw Ahead (bars)	int	200	20 to 500	DrawAheadBars	How far past the current bar an active trendline is drawn. Cosmetic only: the engine keeps the true projected endpoint.
10 NinjaTrader	Debug Mode	bool	false		DebugMode	Writes diagnostics to apx_nt_debug.log in the NinjaTrader 8 folder.
10 NinjaTrader	Dump Bars + Outputs CSV	bool	false		DebugDumpCsv	Writes every committed bar with the engine outputs to apx_dump__.csv (parity checks).

65 parameters.

Sessions, time and volume

Sessions

The indicator has no session logic of its own: every bar the chart delivers is fed to the engine in order, so the chart's Trading Hours template decides which bars exist. With an ETH template (for example `CME US Index Futures ETH`) the overnight bars take part in the pivots, the EMA, the ATR and the volume average; with an RTH template they do not. Both are valid; the values differ, as they do for any indicator. A trendline drawn across a session gap simply spans the bars on both sides, as on the original.

Time stamps

NinjaTrader stamps a bar with its CLOSE time in the platform time zone. The engine's bar open time (used for draw-object tags and alert de-duplication only) is reconstructed as close time minus the bar period; for non-time bars the observed bar spacing is used. Trendlines and level lines are anchored at bar times; the part of an active line that lies past the current bar is positioned on the bar period, so on tick, volume or range bars its x position is approximate.

Volume

`Volume[]` on a futures chart is real contract volume. The original script used `volume`, which on forex feeds is tick volume; on NinjaTrader futures the volume filter therefore compares exchange volume against its own average, a better input for the same test. A bar with zero volume passes the filter, exactly as in the original, and during the first `Volume Average Length` bars the average is not yet seeded, so the filter fails rather than passes.

Contract rollover

A continuous contract (merge policy `MergeBackAdjusted` or `MergeNonBackAdjusted` in the Data Series dialog) shows a price jump at the roll unless back-adjusted. Pivots, trendlines, the breakout margin, the stop and the targets are price differences, so a non-back-adjusted roll can produce a spurious pivot pair and a steep line; with back adjustment there is none. State the merge policy when comparing values across machines.

Compatibility

ITEM	STATUS
NinjaTrader version	8.1.x Desktop (tested 8.1.8.2). Not 8.0, not NinjaTrader Web or mobile.
Product type	Indicator, price panel overlay, single data series.

ITEM	STATUS
Instruments	Any instrument with bars: futures, micro futures, forex, stocks, crypto CFDs. Verified on MNQ (CME) 1-minute; the engine was verified on EURUSD M5 (stage 1 parity).
Bar types	Time-based (second, minute, day, week, month) fully; tick, volume, range, Renko and other non-time bars run, with the alert period string and the drawn line extensions derived from the observed bar spacing.
Calculate modes	OnBarClose (default), OnEachTick, OnPriceChange (forming-bar preview on a scratch copy of the state; signals and alerts on closed bars only). Tick Replay handled as per-tick.
Multi-instance	Yes: every draw object and alert id carries the instrument, period and instance hash.
Market Analyzer	Loads (no chart drawing); the <code>TrendEma</code> , <code>Signal</code> and <code>TradeDirection</code> series are available as columns. Not advertised as a scanner.
Strategy Analyzer / Strategy Builder	The three series can be referenced from a strategy; all inputs are <code>[NinjaScriptProperty]</code> .
Data requirements	Bars only. Volume is used by the volume filter when present; a zero-volume bar passes the filter exactly as in the original (<code>volume == 0</code> counts as a pass), and the filter fails during the volume average's warm-up.
Rendering	Plots and draw objects through the standard NinjaTrader API; the dashboard through SharpDX in <code>OnRender</code> with DPI scaling and full resource cleanup on render-target change and termination.
Dark and light charts	The dashboard uses the original's fixed dark palette on either background.
Templates and workspaces	All settings serialize (colours through the standard brush serializer).
Minimized chart	Keeps calculating and alerting while its chart window is minimized (<code>IsSuspendedWhileInactive</code> off). A bar that closed more than two bar periods ago is never alerted, so a reconnect or a catch-up cannot produce a burst of stale alerts.
Order handling	None. The indicator never touches orders, positions or accounts.

09

Troubleshooting

SYMPTOM	CAUSE AND FIX
No trendlines yet	Every line needs two confirmed pivots, and a pivot is confirmed only <code>Pivot Right Bars</code> bars after it printed. Load more history or lower the pivot lengths.
No EMA line	The EMA needs <code>EMA Length</code> bars (200 by default) before it seeds, or Show EMA on Chart / EMA Trend Filter is off.
Lines but no signals	A signal needs a close through an active line by more than the breakout margin AND <code>Min Confluence Score</code> passing filters AND no same-side signal within the cooldown. Check the FILTERS rows of the dashboard; lower Min Confluence Score to see raw breaks become signals.
Signals print but the TRADE rows stay dashed	Show TP/SL Levels is off. In that state the original script tracks no trade, and this build does the same.

SYMPTOM	CAUSE AND FIX
Dashboard missing	07 Dashboard > Show Dashboard on? The dashboard is drawn in the price panel; with the indicator in its own panel it is not visible.
Dashboard text tiny or huge	Sizes follow the Windows display scaling.
Level price tags cut off at the right edge	They sit twelve bars to the right of the current bar while the trade is open; add right margin (chart Properties > Right side margin).
Trendlines end in empty space to the right	Active lines are drawn Draw Ahead (bars) bars past the current bar; the engine keeps the true projected endpoint regardless.
Alerts do not fire	Alerts fire only for bars that close after the chart loaded, only in realtime, and only with Enable Alerts plus the per-kind toggle on. Email needs the Mail share service configured under Tools > Options > Share.
Sound not played	Sound File must name a file inside <code><NinjaTrader 8 install>\sounds\</code> .
Values differ from another platform on the same symbol	Bars differ by feed and session template; compare on identical bars (see the chapter on how the port was verified).
An error line names Apex Break PRO in the Log tab	Set 10 NinjaTrader > Debug Mode on, reproduce, and send <code>Documents\NinjaTrader 8\apx_nt_debug.log</code> with the chart interval, instrument and NinjaTrader version.
Two dashboards on one chart	Two instances are configured (Ctrl+I, Configured list). Remove one.

10

How the port was verified

The verification has two stages, which measure different things and are reported separately. It is never stated as "verified against TradingView on futures bars": TradingView's CME feed and NinjaTrader's feed do not carry the same bars (sessions, rollover, continuous-contract construction), so there is no such comparison to make.

Stage 1: engine vs the TradingView original (cTrader edition, 8 September 2026)

The platform-free engine that ships here is the same code as the cTrader edition's engine apart from its namespace line. That engine was replayed on 20,561 identical ICMARKETS:EURUSD M5 bars read from the TradingView chart model and compared with a probe copy of the published Pine script:

- 46 scalar series equal on every bar (indicators, pivots, pivot anchors, breakout flags, confluence, cooldown, signals, stored metadata, the whole trade state machine);
- both trendline collections equal across 1,439,270 field comparisons (five slots per side, start and end bar and price, detection bar, projected price, derived slope), the whole-collection integer digest exact on every bar;
- 211 of 211 resistance breaks, 199 of 199 support breaks, 142 of 142 buy signals and 132 of 132 sell signals on the same bars; all ten drawn trendlines equal including solid/dashed style; 275 of 275 signal labels equal in text and anchor; 13 of 15 dashboard rows identical, the other 2 differing by brand text only;
- two parameter sweeps (Show TP/SL Levels off; breakout margin off) also at zero;
- residue: none.

Stage 2: NinjaTrader host vs the engine on identical NinjaTrader bars (16 September 2026)

With **Dump Bars + Outputs CSV** on, the indicator writes every committed bar (time, open, high, low, close, volume) together with the engine outputs as the host saw them, in exactly the columns the cTrader parity harness writes: 119 series covering the indicators, the pivots and their anchors, the breakout flags, the confluence, the cooldown, the signals, the stored signal metadata, the whole trade state machine, and both trendline books (count, active count, active mask, integer digest, and start bar, start price, end bar, end price, detection bar and current line price of five slots per side). The same bars are replayed through a build of the identical engine outside NinjaTrader, with no host in between, and the two files are diffed value by value.

Result on MNQ DEC26, 1-minute bars, NinjaTrader 8.1.8.2, Simulation connection (delayed CME data), Calculate on bar close:

bars compared	5,058
bar identity (time, open, high, low, close)	0 mismatching rows
series compared	119
cells compared	601,902
mismatches	0
max abs difference	0
trendlines created on the run	86 resistance, 87 support
breaks / signals on the run	49 / 49 breaks, 34 buy / 35 sell signals, on the same bars

Second run on the same chart with `Calculate = OnEachTick`, the chart left live for twenty minutes so the dump spans the historical to realtime transition and 23 bars committed through the per-tick `IsFirstTickOfBar` path: 5,080 bars, 119 series, 604,520 cells, 0 mismatches, 0 bar-identity mismatches.

What stage 2 proves: the host feeds the engine exactly one closed bar per chart bar with the right OHLCV, never double-feeds or skips a bar across the historical to realtime transition, and draws the values the engine produced, trendline by trendline. What it does not prove: anything about future performance.

Licensing

License model

The product is distributed as a compiled NinjaScript Add-On and licensed through NinjaTrader's user-based Vendor Licensing (NinjaTrader 8.1.5.0 or later). A licence is tied to the customer's NinjaTrader account login, not to a machine: the same login works on every installation of NinjaTrader Desktop the customer uses, and no Machine ID has to be sent to the vendor. Licences are created in the NinjaTrader Ecosystem Vendor Dashboard against the customer's NinjaTrader account email address, with a licence type and an expiry date.

The indicator carries the `VendorLicense(productId)` call in its constructor, with the Product ID of its add-on in the Vendor Dashboard. NinjaTrader verifies the licence whenever the indicator is used.

No free trial

The product is sold as a paid licence only; no free trial is offered. A licence is issued to the buyer's NinjaTrader account email after payment, normally within a few hours.

Activation and transfer

Nothing to activate: log in to NinjaTrader with the licensed account. A new PC needs no transfer. NinjaTrader allows two simultaneous market-data sessions per account, which is what prevents a licence from being shared.

Behavior without a valid licence

NinjaTrader shows its own licence notice (invalid or expired) and stops calling the indicator's calculation and plot methods: the indicator draws nothing and raises no alerts, while its settings still load. It never places orders (it has no order code) and exposes no proprietary data. A licensing-server outage is handled by NinjaTrader's verification (it retries against a secondary server) and must never crash the chart.

Licensing is a practical access-control layer, not a guarantee against copying.

Support

Questions go through the contact page at <https://kestutisbalciunas.com/contact/> with the NinjaTrader version, the instrument and the chart interval. For an error, switch **11 NinjaTrader > Debug Mode** on, reproduce it once and attach `apx_nt_debug.log` from the `Documents/NinjaTrader 8` folder; switch Debug Mode off afterwards.

The product page, with the documentation and the change log, is <https://kestutisbalciunas.com/ninjatrader/apex-break-pro/>. Updates and refunds follow the terms of the NinjaTrader listing once it is live.

13

Risk disclosure

Trading involves substantial risk. Past, simulated, and backtested performance is not necessarily indicative of future results.

Futures, foreign currency and options trading contains substantial risk and is not for every investor. An investor could potentially lose all or more than the initial investment. Risk capital is money that can be lost without jeopardizing one's financial security or lifestyle. Only risk capital should be used for trading and only those with sufficient risk capital should consider trading.

Apex Break PRO is a technical analysis tool. It draws trendlines, levels and signals computed from historical bars; it does not place, modify or close orders and it gives no trade advice. The star rating describes how many of the enabled filters agreed on the breakout bar; it is a description of agreement, not a probability, and not a forecast. Nothing in the software or its documentation is a promise of returns, win rates or funded-account success.

NinjaTrader is a registered trademark of NinjaTrader Group, LLC. No NinjaTrader company has any affiliation with the owner, developer, or provider of the products or services described herein, or any interest, ownership or otherwise, in any such product or service, or endorses, recommends or approves any such product or service.