Kestutis Balciunas

NINJATRADER 8 INDICATOR · NINJASCRIPT, C#

User Guide

An adaptive Ichimoku cloud with five signal families, a confluence score, a stop and three targets, multi-timeframe rows, a live dashboard and alerts.

PRODUCT	Kumo Forge Pro for NinjaTrader 8
VERSION	1.0.0
DATE	November 2026
NINJATRADER	8.1.x Desktop, tested on 8.1.9.0
PACKAGE	`KumoForgePro.zip`, compiled NinjaScript Add-On
SHA-256	`37f521e3cd3069a841d6ea8adc16d8b88c90e5312694fab04d5ca7fcd7a34ff5`
AUTHOR	Kestutis Balciunas, https://kestutisbalciunas.com/
PRODUCT PAGE	https://kestutisbalciunas.com/ninjatrader/kumo-forge-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 this edition was verified

11 Licensing

12 Support

13 Risk disclosure

01

What it is

Chart indicator for NinjaTrader 8 Desktop (NinjaScript, C#): an Ichimoku cloud whose periods can scale with volatility, five signal families evaluated on closed bars, a confluence score out of 100 (110 with the RSI factor), a stop and three targets for the newest signal, zones where a flat Kijun ends, three higher-timeframe rows, a dashboard and alerts. It only draws and alerts: it never places, modifies or closes orders.

02

Installation

Requirements

- NinjaTrader 8.1.x Desktop for Windows (tested on 8.1.9.0). Not for NinjaTrader 8.0 or NinjaTrader Web.
- Any data connection that delivers bars for the chart's instrument, including minute bars for the three higher timeframes. No Order Flow+ features, no depth, no tick history.

Import the Add-On package

1. Control Center > Tools > Import > NinjaScript Add-On.
2. Select `KumoForgePro.zip` and confirm. NinjaTrader compiles the assembly and reports that the import is 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 with at least a few hundred bars (any instrument, a 1 to 60 minute interval is a good first look), press Ctrl+I.
2. In the Available list find **Kumo Forge Pro**, click add, click OK.
3. The chart shows the green or red cloud, the blue Tenkan-Sen, the pink Kijun-Sen, the orange Chikou span, the dashboard in the top-right corner and the product name at the bottom. Signal plates and markers appear from about the 80th bar on, where the filters let them through.

The indicator loads three extra data series, 500 bars each at the defaults. On a first load the data may take a moment to arrive.

Uninstall / update

- Remove: Control Center > Tools > Remove NinjaScript Assembly > Kumo Forge 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.

03

Quick start

First chart

1. Add **Kumo Forge Pro** to a chart (Ctrl+I). Leave every setting at its default.
2. No signal is allowed on the first bars: the indicator needs Span B Period + Displacement bars, at least 80. Give the chart some history.

Reading the chart

- **Cloud (Kumo)**: the band between Senkou A and Senkou B, green where A is on top and red where B is. It is plotted ahead of the bars that computed it, so its tip reaches past the last bar. Chart Properties > Right side margin makes room to see it.
- **Blue line, pink line**: the Tenkan-Sen and the Kijun-Sen. In the default Adaptive mode their periods scale with volatility.
- **Orange line**: the Chikou span, plotted behind the bar that computed it.
- **Long and Short plates**: a Tenkan/Kijun cross that passed the filters, with its score.
- **KB, CB**: a bounce off the Kijun or off a cloud edge, in a trend already running.
- **BK Up, BK Dn, a circle**: a close through the cloud on above-average volume, and a return to the broken edge within ten bars.
- **Cross marker**: the Chikou span crossing past price. Off by default.
- **Lines with price tags**: the entry, the stop and three targets of the newest signal. They are rebuilt on every new bar. A tag past the last bar is held just inside the chart.
- **Pink zones**: where a flat Kijun ended, a level the line sat on.
- **Dashboard**: the market data, the Ichimoku state, the last signal and its score, the tracked trade, and the three higher-timeframe rows.

Settings to try first

SETTING	WHAT IT CHANGES
Min Confluence Score	The one setting that changes the number of signals most. 60 to 70 is a noticeable tightening.
Calculation Mode	Classic uses fixed periods. Adaptive scales them with volatility.
Volume Filter, **Chikou Confirmation**	Off lets more signals through.
SL Mode, **SL ATR Buffer**	Where the stop sits. ATR mode uses two ATRs and ignores the buffer.
TF 1, TF 2, TF 3	The three higher timeframes of the panel. Above a 240-minute chart, raise them.
Position, **Font Size**	The dashboard's corner and text size.
Calculate (NinjaTrader's own setting)	OnBarClose updates once per closed bar. OnEachTick follows the forming bar too; signals still wait for the close.

Alerts

Alerts appear in the Alerts Log window on a live chart. Group **13 NinjaTrader Alerts** has one switch per event, all on by default, plus the sound and e-mail channels, which are off by default.

What it does not do

It places no orders. The score, the trade lines and the win-rate row describe what the bars on this chart did at these settings; none of them is a forecast or a trading record.

04

Calculate mode and bar timing

- Default `Calculate.OnBarClose`: one commit per closed bar.
- `OnEachTick` / `OnPriceChange`: the closed bar is committed once, at the first tick of the next bar, with the same result. The forming bar is previewed against a scratch copy of the engine's state, so the plots and the dashboard follow the market and nothing of a preview stays behind. The surface refreshes about four times a second.
- A still-forming last historical bar is previewed, not committed. A bar's OPEN time is its NinjaTrader close time minus the bar period; a shortened session bar opens at the previous bar's close.
- **Every signal belongs to a closed bar and never moves.** The plates, markers and zones of a closed bar stay where they are.
- **The trade lines and their tags do move.** The five lines are rebuilt on the newest bar so they keep reaching past it, and their left end follows the bar of the newest signal, at most 100 bars back.
- **The higher-timeframe rows read closed bars only.** A chart bar receives the state of the newest higher-timeframe bar that had closed by the chart bar's own close, and a bar still forming is never read. With OnBarClose a row can change only when a chart bar closes; with OnEachTick it changes when the higher-timeframe bar closes. The rows are not a condition of any signal.
- Alerts are raised in real time only, once per kind and bar, and never for a bar older than two bar periods.

05

Outputs for strategies and the Market Analyzer

PLOT / SERIES	VALUES
`TenkanSen` (plot 0, line)	the conversion line; empty with Show TK Lines off
`KijunSen` (plot 1, line)	the base line; empty with Show TK Lines off
`ChikouSpan` (plot 2, line)	the lagging span, on the bar it is plotted at, so the newest bars carry no value yet; empty with Show Chikou Span off

PLOT / SERIES	VALUES
`SenkouA` (plot 3, hidden)	the cloud's first edge on this bar; empty with Show Cloud off
`SenkouB` (plot 4, hidden)	the cloud's second edge on this bar; empty with Show Cloud off
`Signal` (plot 5, hidden)	+1 on a closed bar with any buy signal, -1 with any sell signal, 0 otherwise

The series read at bar close. On a forming bar under OnEachTick or OnPriceChange, `Signal` reads 0 until the bar has closed.

06

Parameters

The inputs keep the titles, order, defaults and ranges of the other editions (the group names carry a number prefix because NinjaTrader sorts groups alphabetically). Groups 12 and 13 hold the NinjaTrader settings, which change no calculation. The indicator makes no network call, so Webhook JSON Format selects the alert text only.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
01 Main Settings	Calculation Mode	KfpMode	Adaptive		`Mode`	Classic: fixed Ichimoku periods. Adaptive: periods scale with volatility, faster in volatile markets, slower in quiet ones.
01 Main Settings	Conversion Period	int	9	3 to 50	`ConversionPeriod`	Tenkan-Sen lookback. In Adaptive mode it scales with Adapt Strength. Lower is faster and noisier.
01 Main Settings	Base Period	int	26	10 to 100	`BasePeriod`	Kijun-Sen lookback. In Adaptive mode it scales with volatility.
01 Main Settings	Span B Period	int	52	20 to 200	`SpanBPeriod`	Senkou Span B lookback. Longer gives a thicker cloud.
01 Main Settings	Displacement	int	26	1 to 100	`SpanDisplacement`	Cloud and Chikou displacement in bars. Both shift by this value minus one.
01 Main Settings	Source	KfpSource	Close		`Source`	The price the Chikou span and the RSI read from each bar. NinjaTrader's Input series row has no effect: use this setting.
02 Adaptive Engine	Volatility Lookback	int	50	10 to 200	`VolatilityLookback`	ATR window for the adaptive scaling. Higher adapts more smoothly.
02 Adaptive Engine	Adapt Strength	double	0.4	0.1 to 0.8	`AdaptStrength`	How far the periods stretch: 0.1 is plus or minus 10 percent, 0.8 is plus or minus 80 percent.
03 Signal Types	Kijun Bounce Signals	bool	true		`ShowKijunBounce`	Signals when price bounces off the Kijun in a trend: an uptrend for a buy, a downtrend for a sell.
03 Signal Types	Cloud Edge Bounce Signals	bool	true		`ShowCloudBounce`	Signals when price bounces off a cloud edge: above the cloud for a buy, below it for a sell.
03 Signal Types	Chikou Breakout Signals	bool	false		`ShowChikouBreak`	Signals on the Chikou span crossing past price. Off by default: it is the most delayed confirmation.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
03 Signal Types	RSI Confluence Filter	bool	false		`RsiFilter`	Adds RSI(14) alignment as a score factor: above 50 for a buy, below 50 for a sell, 10 points.
04 Filters	Min Confluence Score	int	40	0 to 110	`MinConfluenceScore`	The score a signal needs. Higher gives fewer signals. Above 100 needs RSI Confluence Filter on.
04 Filters	Kumo Thickness Filter	bool	true		`KumoFilter`	Suppresses signals in very thin parts of the cloud.
04 Filters	Volume Filter	bool	true		`VolumeFilter`	Requires volume above its 20-bar average by a fifth. A bar with no volume passes.
04 Filters	Chikou Confirmation	bool	true		`ChikouFilter`	Requires the Chikou span to agree with the signal's direction.
05 Smart TP/SL	Show TP/SL Lines	bool	true		`ShowTPSL`	Tracks a trade after each signal and draws its levels. Off, no trade is tracked at all: the dashboard reads No Active and alert bodies print zero levels.
05 Smart TP/SL	SL Mode	KfpSlMode	KumoEdge		`SlMode`	KumoEdge: the far cloud edge plus the buffer. Kijun: the Kijun plus the buffer. ATR: two ATRs from the entry, ignoring the buffer.
05 Smart TP/SL	SL ATR Buffer	double	0.5	0.1 to 3.0	`SlAtrBuffer`	ATR multiple added beyond the stop level in KumoEdge and Kijun modes.
05 Smart TP/SL	TP1 R:R	double	1.5	0.5 to 5.0	`Tp1RR`	Target 1 as a multiple of the stop distance. The distance is floored at half an ATR.
05 Smart TP/SL	TP2 R:R	double	2.5	1.0 to 8.0	`Tp2RR`	Target 2 as a multiple of the stop distance.
05 Smart TP/SL	TP3 R:R	double	4.0	1.5 to 12.0	`Tp3RR`	Target 3 as a multiple of the stop distance. Reaching it closes the tracked trade.
06 Kumo Breakout	Show Kumo Breakouts	bool	true		`ShowBreakout`	Signals when price closes through the cloud from the other side.
06 Kumo Breakout	Breakout Vol Mult	double	1.3	1.0 to 3.0	`BreakoutVolMult`	Volume must exceed its 20-bar average by this multiple for a breakout.
06 Kumo Breakout	Show Retest Signals	bool	true		`ShowRetest`	Signals when price returns to the cloud edge within ten bars of a breakout and closes back on the breakout side.
07 Multi-Timeframe	Show MTF Panel	bool	true		`ShowMTF`	Shows the three higher-timeframe rows on the dashboard. Off, the panel is shorter and the alignment alerts still run.
07 Multi-Timeframe	TF 1	KfpTimeframe	Minute15		`MtfTf1`	First timeframe of the panel. Loads a data series of the chart's instrument.
07 Multi-Timeframe	TF 2	KfpTimeframe	Hour1		`MtfTf2`	Second timeframe of the panel.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
07 Multi-Timeframe	TF 3	KfpTimeframe	Hour4		`MtfTf3`	Third timeframe of the panel.
08 Visual Settings	Theme	KfpTheme	Auto		`Theme`	Dashboard colours. Auto reads the chart background. On Light the section titles are white on pale blue: use Dark for a legible panel on a light chart.
08 Visual Settings	Show Cloud	bool	true		`ShowCloud`	Draws the Kumo fill.
08 Visual Settings	Show TK Lines	bool	true		`ShowTK`	Draws the Tenkan-Sen and the Kijun-Sen.
08 Visual Settings	Show Chikou Span	bool	true		`ShowChikou`	Draws the lagging span.
08 Visual Settings	Show TK Signal Labels	bool	true		`ShowSignals`	Draws the Long and Short plates on Tenkan/Kijun cross signals.
08 Visual Settings	Show Score on Signals	bool	true		`ShowScore`	Adds the confluence score to the plates.
08 Visual Settings	Show Breakout Labels	bool	true		`ShowBreakLabels`	Draws the breakout and retest markers.
08 Visual Settings	Show TP/SL Labels	bool	true		`ShowTPSLLabels`	Draws the price tags beside the entry, stop and target lines.
08 Visual Settings	Show S/R Zones	bool	true		`ShowZones`	Draws a zone where a flat Kijun ends.
08 Visual Settings	Show Trend Background	bool	false		`ShowBg`	Tints the background of bars in a bullish or bearish trend state.
08 Visual Settings	Signal Label Size	KfpSize	Small		`SignalLabelSize`	Text size of the Long and Short plates.
08 Visual Settings	Show Watermark	bool	true		`ShowWatermark`	Draws the product name at the bottom of the chart.
09 Dashboard	Show Dashboard	bool	true		`ShowDashboard`	Draws the dashboard. Nothing calculated depends on it.
09 Dashboard	Position	KfpDashPos	TopRight		`DashPosition`	The chart corner the dashboard sits in.
09 Dashboard	Font Size	KfpSize	Normal		`DashFontSize`	Text size of the dashboard.
10 Alerts	Webhook JSON Format	bool	false		`WebhookJson`	Writes alert texts as JSON. The text goes to the Alerts Log, the Output window and the e-mail: this edition makes no network call and posts nothing.
11 Colors	Bull	colour	#00E676		`BullColor`	Buy plates, target lines and the cloud where Senkou A is on top.
11 Colors	Bear	colour	#FF5252		`BearColor`	Sell plates, the stop line and the cloud where Senkou B is on top.
11 Colors	Tenkan	colour	#2962FF		`TenkanColor`	The Tenkan-Sen line. This setting decides the colour, over the Plots section.
11 Colors	Kijun	colour	#E91E63		`KijunColor`	The Kijun-Sen line and the flat-Kijun zones. This setting decides the line colour, over the Plots section.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
11 Colors	Chikou	colour	#FF9800		`ChikouColor`	The Chikou span and its markers. This setting decides the line colour, over the Plots section.
11 Colors	Breakout	colour	#00BCD4		`BreakoutColor`	Breakout, retest and cloud-bounce markers.
12 NinjaTrader	Label Glyphs	KfpGlyphs	Unicode		`Glyphs`	Unicode draws the dots, dashes and marker shapes as symbols. Ascii uses plain characters, if a font on your machine shows boxes.
12 NinjaTrader	Signal Label Cap	int	450	10 to 500	`SignalLabelCap`	How many plates are kept before the oldest is dropped.
12 NinjaTrader	S/R Box Cap	int	180	10 to 200	`SrBoxCap`	How many flat-Kijun zones are kept.
12 NinjaTrader	Max Visual History (bars, 0 = all)	int	0	0,int.MaxValue	`MaxVisualHistory`	Limits how far back the markers, plates and zones are drawn. Changes nothing that is calculated.
12 NinjaTrader	Debug Mode	bool	false		`DebugMode`	Writes diagnostics to kfp_nt_debug.log in the NinjaTrader 8 folder.
12 NinjaTrader	Dump Bars + Outputs CSV	bool	false		`DebugDumpCsv`	Writes every committed bar with the engine outputs to kfp_dump_<instrument>_<period>.csv and the higher-timeframe bars to kfp_altbars files (parity checks).
13 NinjaTrader Alerts	Alert: TK Cross	bool	true		`AlertTkCross`	The Tenkan and Kijun crossing with a signal, both directions.
13 NinjaTrader Alerts	Alert: Kumo Breakout	bool	true		`AlertKumoBreak`	Price closing through the cloud, both directions.
13 NinjaTrader Alerts	Alert: Retest	bool	true		`AlertRetest`	Price returning to the cloud edge after a breakout, both directions.
13 NinjaTrader Alerts	Alert: Kijun Bounce	bool	true		`AlertKijunBounce`	A bounce off the Kijun, both directions.
13 NinjaTrader Alerts	Alert: Cloud Bounce	bool	true		`AlertCloudBounce`	A bounce off a cloud edge, both directions.
13 NinjaTrader Alerts	Alert: Chikou Break	bool	true		`AlertChikouBreak`	The Chikou span crossing past price, both directions.
13 NinjaTrader Alerts	Alert: TP1 Hit	bool	true		`AlertTp1`	The first target being reached.
13 NinjaTrader Alerts	Alert: TP2 Hit	bool	true		`AlertTp2`	The second target being reached.
13 NinjaTrader Alerts	Alert: TP3 Hit	bool	true		`AlertTp3`	The third target being reached, which also closes the tracked trade.
13 NinjaTrader Alerts	Alert: SL Hit	bool	true		`AlertSl`	The stop being reached.
13 NinjaTrader Alerts	Alert: MTF Alignment	bool	true		`AlertMtfAlign`	All three higher timeframes agreeing, in either direction.
13 NinjaTrader Alerts	Alert: Trend Change	bool	true		`AlertTrendChange`	The trend state becoming bullish, bearish or neutral.
13 NinjaTrader Alerts	Alert: Strong Signal	bool	true		`AlertStrong`	A cross whose confluence score is 80 or more.
13 NinjaTrader Alerts	Alert Log Line	bool	true		`AlertLogLine`	Writes each alert to the NinjaScript Output window. Every alert also appears in the Alerts Log window.

GROUP	PARAMETER	TYPE	DEFAULT	RANGE	PROPERTY	NOTES
13 NinjaTrader Alerts	Alert Sound	bool	false		`AlertSound`	Plays a sound with each alert.
13 NinjaTrader Alerts	Alert Sound File	string	"Alert2.wav"		`AlertSoundFile`	A file name inside the NinjaTrader sounds folder, for example Alert2.wav, or the full path to a .wav file. Blank plays Alert2.wav.
13 NinjaTrader Alerts	Alert E-mail	bool	false		`AlertEmail`	Sends each alert by e-mail through the Mail share service configured in NinjaTrader (Tools > Options > Share).
13 NinjaTrader Alerts	E-mail To	string	""		`EmailTo`	Recipient address for alert e-mails. Blank sends none.

75 parameters.

07

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 Ichimoku windows, the ATR and the volume average; with an RTH template they do not. Both are valid, and the values differ, as they do for any indicator.

Time stamps

NinjaTrader stamps a bar with its CLOSE time in the platform time zone. The engine's bar open time (used for the alert de-duplication and the CSV dump) is reconstructed as close time minus the bar period and converted to UTC. A shortened session bar opens at the previous bar's close. A day-based bar is stamped with a date and no time, so it opens at the session end of the bar before it. On non-time bars a bar opens at the previous bar's close.

The lines and zones that reach past the last bar are positioned on the chart's bar spacing, so on a chart with a gap ahead their x position is approximate.

The three extra series

TF 1, TF 2 and TF 3 (15, 60 and 240 minutes by default) read series of the chart's instrument, built by NinjaTrader from the same data and the chart's Trading Hours template. Each one loads its own number of bars rather than the chart's date range: 500 at the defaults, because a 5-day chart would hold 30 bars of 240 minutes, too few for a 52-period Span B. Until a window has filled, its span falls back to the bar's own midpoint.

Each row is a classic Ichimoku test on that timeframe's own bars: **Bull** when the close is above both spans and the fast line above the slow one, **Bear** when the close is below both and the fast line below the slow one, **Neutral** otherwise. It always uses the fixed periods and the close, whatever Calculation Mode and Source say.

A row reads closed bars only. A chart bar receives the state of the newest higher-timeframe bar that had closed by the chart bar's own close, and a bar still forming is never read, on history or live. A daily, weekly or monthly bar counts as closed at the session end of its date. A row can therefore be one bar behind a bar that has just ended on a closed market, until the next session's first tick.

The rows drive the dashboard and the two alignment alerts. They are not a condition of any signal.

Volume

Volume on a futures chart is real contract volume. On a forex feed it is usually tick volume. The volume filter (above the 20-bar average by a fifth), one score factor (10 points) and the breakout test (above the average by **Breakout Vol Mult**) all compare a bar with its own instrument's recent bars. A bar with no volume passes those tests, and the dashboard's Volume row reads N/A.

A live bar is built from ticks, and after a reload the same bar comes from the provider's history. Their volumes can differ slightly, so on a volume-sensitive bar a signal seen live can differ after a reload. This is the data, not the calculation.

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. The Ichimoku windows and the ATR span that jump, so a non-back-adjusted roll widens the cloud and the stop distance until it has left the windows. State the merge policy when comparing values across machines.

08

Compatibility

ITEM	STATUS
NinjaTrader version	8.1.x Desktop (tested 8.1.9.0). Not 8.0, not NinjaTrader Web or mobile.
Product type	Indicator, price panel overlay. Multi-series: the chart's own bars plus three series of the same instrument, used by the dashboard's higher-timeframe rows and the two alignment alerts only.
Instruments	Any instrument with bars: futures, micro futures, forex, stocks.
Bar types	Time-based (second, minute, day, week, month) fully. Tick, volume, range, Renko and other non-time bars run; on them the part of the lines past the last bar is positioned on the observed bar spacing, and a higher-timeframe row reads the bars that closed by the chart bar's close.
Calculate modes	OnBarClose (default), OnEachTick, OnPriceChange (a forming-bar preview that leaves nothing behind; signals on closed bars only). Tick Replay is handled as per-tick.
Multi-instance	Yes. Every instance paints its own surface, and alert ids carry the instrument, the period and the instance.
Market Analyzer	Loads with no drawing; `TenkanSen`, `KijunSen`, `ChikouSpan`, `SenkouA`, `SenkouB` and `Signal` are available as columns. Not advertised as a scanner.
Strategies	The six series can be read from a strategy; the settings are `[NinjaScriptProperty]`. A strategy hosting the indicator must add the same three data series itself, with the same timeframes.
Data requirements	Bars only. Volume is used by the volume filter, one score factor and the breakout test; a bar with no volume passes those tests. Each higher timeframe loads 500 bars by default (150 weekly, 120 monthly, more if Span B Period needs it).
Rendering	SharpDX in `OnRender` with DPI scaling and full resource cleanup on render-target change and termination. No draw objects. The price scale follows the candles, so a cloud far from price can leave the panel.
Dark and light charts	Theme = Auto reads the chart background. On the light colours the dashboard's section titles are white on pale blue, in every edition: choose Theme = Dark for a legible panel on a light chart.
Templates and workspaces	All settings serialize (colours through the standard brush serializer).

ITEM	STATUS
Minimized chart	Keeps calculating and alerting while its chart window is minimized (`IsSuspendedWhileInactive` off). A bar that closed more than two bar periods ago is never alerted, so a reconnect never replays old alerts.
Network	None. The indicator opens no connection. An e-mail goes through NinjaTrader's own Mail share service.
Order handling	None. The indicator never touches orders, positions or accounts.

09

Troubleshooting

SYMPTOM	CAUSE AND FIX
Nothing on the chart	Check that the licence is active for your NinjaTrader login and that the indicator is in the Configured list (Ctrl+I). It paints in the price panel: with the indicator in its own panel it is not visible.
The first bars have no signals	A signal needs Span B Period + Displacement bars, at least 80. Load more bars.
Signals are too rare or too frequent	**Min Confluence Score** changes this most: 60 to 70 is a noticeable tightening. For more signals lower it, and consider switching **Volume Filter** and **Chikou Confirmation** off.
A score above 100 never fires	The ninth factor is the only one that pays above 100, and it needs **RSI Confluence Filter** on.
The alert text prints a zero stop and zero targets	**Show TP/SL Lines** is off. It is more than a display switch: with it off no trade is tracked, so there are no levels to print. Switch it on.
The dashboard's Status stays No Active	The same switch, or no signal has fired yet.
Turning SL ATR Buffer up moves nothing	SL Mode is ATR, which uses two ATRs and ignores the buffer. Use KumoEdge or Kijun.
The R:R disagrees with the drawn levels	The distance the targets are measured from is floored at half an ATR. A stop closer than that gives targets measured from the floor. Widen the stop.
A trade shows SL X Closed on a bar that reached the third target	A bar wide enough to reach both sets both flags, and the stop is read first.
The Align row is missing from the dashboard	The Version row is written over it, in every edition. The alignment is still computed, still drives its two alerts, and can be read from the three rows above. With Show MTF Panel off, the Version row replaces Bars Since in the same way.
The Cloud row says Thin and gives no direction	Thin takes precedence over Above and Below. The Trend row still gives the direction.
A higher-timeframe row reads Neutral for a long time	A row needs closed bars of that timeframe. Check the Output window and, with Debug Mode on, `kfp_nt_debug.log` : the `feed` lines show how many bars loaded. A data feed short of history gives a row built on a thin window.
A higher-timeframe row did not change at that timeframe's close	With OnBarClose a row changes when a chart bar closes. On a chart slower than the row's timeframe that can be later. OnEachTick updates it sooner.

SYMPTOM	CAUSE AND FIX
The trade lines moved	They are rebuilt on every new bar so they keep reaching past it, and the left end follows the newest signal. Signals never move once their bar has closed.
The cloud's tip past the last bar is not visible	Add room with chart Properties > Right side margin.
The cloud leaves the top or bottom of the panel	The price scale follows the candles. Drag the price axis to make room.
Colours in the Plots section do nothing	**11 Colors** decides the Tenkan, Kijun and Chikou colours. The Plots section still sets width and style.
The Input series row does nothing	The indicator reads each bar's prices through **Source**.
NinjaTrader's own Displacement row shifts the plots away from the cloud	Leave that row, in the indicator's Visual section, at 0. The product's **Displacement** is in group 01.
The section titles of the dashboard are hard to read	On the light colours they are white on pale blue, in every edition. Choose Theme = Dark.
The dashboard shows boxes instead of dots, or a marker shows a box	A font on your machine lacks the character. Set **12 NinjaTrader > Label Glyphs** to Ascii.
No alerts appear	Alerts need a live chart: nothing fires while history loads. A bar that closed more than two bar periods ago is skipped. Check the switch in group 13.
No alert e-mail arrives	Set **Alert E-mail** and **E-mail To**, and set up the Mail share service in Tools > Options > Share.
A webhook receiver gets nothing	This edition makes no network call and posts nothing. **Webhook JSON Format** only changes the text in the Alerts Log, the Output window and the e-mail.
The JSON text is rejected	Eight of the 22 bodies are not valid JSON, in every edition: TP1, TP2, TP3, SL, Kijun Bounce buy and sell, Cloud Bounce buy and sell. Each ends in an unquoted number and a stray quote, like `"price":1.16343"}`. Remove the `"` before the final `}`.
Panning is slow on a very long history	Set **Max Visual History** to something like 5000. It limits how far back markers, plates and zones are drawn and changes nothing that is calculated.
Values differ from another platform on the same symbol	Bars and volume differ by feed and session template, and the volume tests gate most signals; compare on identical bars (see the chapter on how the port was verified).
Text is tiny or huge	Sizes follow the Windows display scaling, **Font Size** and **Signal Label Size**.
An error line names Kumo Forge Pro in the Log tab	Set **12 NinjaTrader > Debug Mode** on, reproduce it, and send `Documents\NinjaTrader 8\kfp_nt_debug.log` with the chart interval, instrument and NinjaTrader version.
Two dashboards on one chart	Two instances are configured (Ctrl+I, Configured list). Remove one.

A daily row, a VWAP or a previous-day level looks wrong or stops updating

NinjaTrader can keep using an older copy of its daily or minute history on your PC, and reloading the chart does not refresh it. Open **Control Center > Tools > Historical Data**, expand **Download**, pick the instrument, select **Day**, set the dates to cover the missing days and press **Download**. Do the same with **Minute** if intraday bars are missing, then reload the chart.

10

How this edition was verified

The verification has two stages, which measure different things and are reported separately.

Stage 1: the engine

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 bar by bar there on EURUSD M5: 15,026,434 comparisons on 20,090 bars across 8 probe runs and 15 input sweeps, plus a comparison of the drawn objects. Every series the product computes, draws or alerts on is at 0 mismatches. 94 differences remain, all in diagnostic time columns that nothing in the indicator reads: they sit on bars next to a session end, where bar data cannot say when a shortened session bar closed.

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

With **Dump Bars + Outputs CSV** on, the indicator writes the bars it saw (time, open, high, low, close, volume) together with the engine outputs as the host received them, 212 series under the same column names, and the higher-timeframe bars behind them. The same bars are replayed through a build of the identical engine outside NinjaTrader, with no host in between, and every column is compared.

RUN	BARS	SERIES	CELLS	MISMATCHES
MNQ 1 Minute, historical load	4,923	212	1,043,676	0
MNQ 1 Minute, across the switch from history to live bars, Calculate = OnEachTick	4,213	212	893,156	0

Both runs also found 0 bar-identity mismatches. The live run had about 20 bars closing live.

11

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.

12

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 **12 NinjaTrader > Debug Mode** on, reproduce it once and attach `kfp_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/kumo-forge-pro/. Updates and refunds follow the terms on that page.

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.

Kumo Forge Pro is a technical analysis tool. It draws an Ichimoku cloud, signals and levels computed from historical bars; it does not place, modify or close orders and it gives no trade advice. The confluence score, the grade and the dashboard's win-rate row describe what the bars on the chart did at the chosen settings; they are descriptions, not probabilities and not forecasts. 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.