NINJATRADER 8 - FULL C# SOURCE

NSL TrendCross Blueprint

User & Developer Guide

Installing, configuring and editing the strategy

Version 1.0 - NinjaTrader 8 (8.1.x) - Windows

No DLL, no activation, no expiry. One plain .cs file.

Support: the developer who wrote it.

ninjastrategyloader.com

Contents

Read this first

This product is a working strategy **and** a code chassis. Sections 1 to 5 are for running it. Sections 6 and 7 are for changing it. If you never intend to open the file in an editor, you can stop after Section 5 — but the value of what you bought is mostly in Section 6.

SECTION 1

What you bought

A complete NinjaTrader 8 automated strategy, delivered as one plain `.cs` file of roughly 2,400 lines of commented C#. There is no DLL, no compiled assembly, no activation server, no licence check and no expiry date. You can open the file in Notepad and read every line before you compile it.

The signal is the small part

Out of the box the strategy enters on a fast/slow moving-average cross. That logic is about forty lines. The remaining 2,360 lines are the machinery that every automated strategy needs and that takes weeks to get right:

- Bracket order tracking through real `Order` objects
- Break-even and auto-trail stop management
- An Exit & Reverse state machine
- Daily profit target and max-loss cut-off
- Chart Trader buttons wired into the NinjaTrader UI
- A floating live-settings window
- A draggable on-chart info panel drawn with SharpDX

Replace the forty-line signal with your own idea and all of the above keeps working unchanged. That is the point of the product.

> **An honest word about the moving-average cross**
>
> A fast/slow MA cross is a public, well-studied entry method. It is not a secret and it is not an edge. Like every MA cross ever written, it whipsaws in ranging markets. No profitability is claimed for it anywhere in this product. What you are buying is a correct, tested execution chassis so that whatever signal you eventually trust runs properly from the first day. Test on simulation before risking capital.

What is in the download

Item	Notes
`NSL_TrendCross.cs`	The complete source. Everything is in this one file.
This guide	Installation, settings reference, and the developer sections.
Demo video	42 seconds showing the strategy running and the live settings window.

Requirements

Requirement	Detail
NinjaTrader 8	Any 8.1.x release. NinjaTrader 7 is not supported.
Operating system	Windows. The settings window uses WinForms.
Dependencies	None. No NuGet packages, no external DLLs.
Market data	Any instrument NinjaTrader 8 can trade.
Skill level	None to run it. Basic C# to change the signal.

SECTION 2

Installing and first run

Import the file

1. In NinjaTrader, open **New** → **NinjaScript Editor**.
2. Choose **File** → **Import** and select `NSL_TrendCross.cs`. Alternatively, copy the file directly into `Documents\NinjaTrader 8\bin\Custom\Strategies\`.
3. Press **F5** to compile. You should see zero errors and zero warnings.

> **If compilation fails**
>
> The most common cause is a second copy of the file already on the machine, which produces duplicate-type errors on the enums. Delete the older copy and compile again. If it still will not compile on a supported NinjaTrader 8 installation, contact support — that is the one case covered by a full refund.

Attach it to a chart

1. Open a chart for the instrument you want to trade.
2. Right-click the chart and choose **Strategies**.
3. Select **NSL_TrendCross** from the list and click **Add**.
4. Set **Quantity**, **Target** and **Stop Loss** for your instrument. These are in **ticks**, not points and not currency.
5. Choose the account. Use **Sim101** until you have watched it trade for a full session.
6. Tick **Enabled** and click **OK**.

First-run checklist

Check	What you should see
Chart Trader panel	Four new controls: AUTO TRADE, LONG, SHORT and SETTINGS.
Info panel	A gold-headed card near the top-left of the chart showing TREND.
Moving averages	Two lines plotted on the chart, violet and pink by default.
Output window	No exceptions. Any error text appears here first.

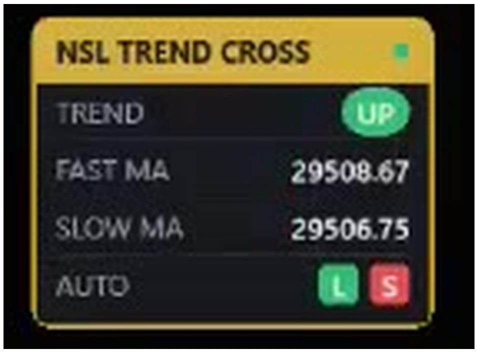
> **Always start on simulation**
>
> Run the strategy on a Sim account for at least one full session before connecting it to a live account. This applies to the unmodified strategy and doubly to any version you have edited. An automated strategy can lose money in ways manual trading cannot, including during platform outages, disconnects and fast markets.

SECTION 3

The interface

Three pieces of user interface are added to the chart: an info panel, a set of Chart Trader buttons, and a floating settings window.

3.1 The info panel

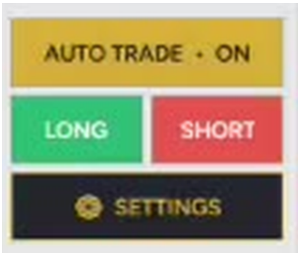


The on-chart info panel. Click and drag it anywhere on the chart.

Row	Meaning
TREND	UP when the fast MA is above the slow MA, DOWN when it is below.
FAST MA	Current value of the fast moving average.
SLOW MA	Current value of the slow moving average.
AUTO	L and S indicators showing whether long and short entries are permitted.

The panel is drawn with SharpDX, the same renderer NinjaTrader uses internally, so it stays responsive on a chart loaded with data. Turn it off with **Show Info Panel?** in the strategy properties.

3.2 Chart Trader buttons



Four controls added to the Chart Trader panel.

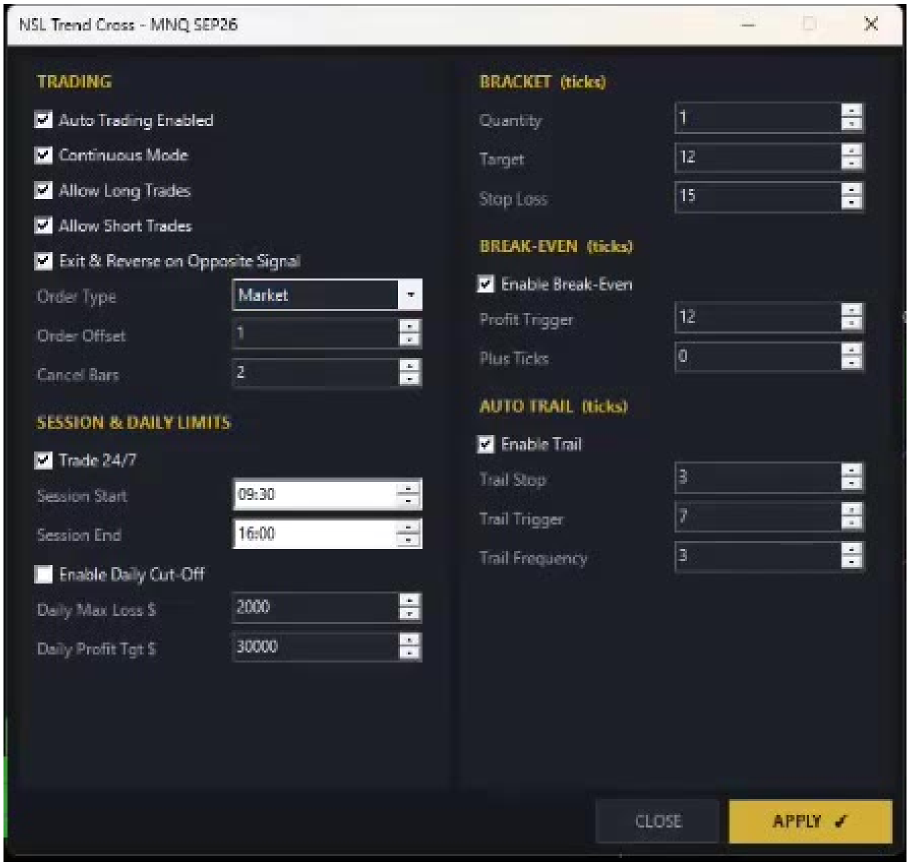
Button	What it does
AUTO TRADE	Master on/off switch. When OFF, no new entries are taken. Existing positions and their brackets are left alone.

Button	What it does
LONG	Toggles permission for long entries. Turn it off to trade short only.
SHORT	Toggles permission for short entries. Turn it off to trade long only.
SETTINGS	Opens the live settings window described below.

Turning AUTO TRADE off does not flatten you

It stops new entries. If you are in a position when you press it, that position and its stop and target stay working. To exit immediately, use the Chart Trader **Close** button or flatten from the Control Center.

3.3 The live settings window



The settings window, opened from the SETTINGS button.

This window is the feature most people buy the chassis for. It changes the strategy's settings **while the strategy is running and while you hold a position**. The standard NinjaTrader properties dialog cannot do this — changing a property there requires disabling the strategy, which closes your position tracking.

Edit the values you want, then press **APPLY**. The new values take effect on the next bar. **CLOSE** dismisses the window without applying anything.

> **What APPLY does and does not change**
>
> APPLY changes the settings used for **future** entries and for ongoing stop management. It does not retroactively move the target or stop of a position that is already open unless the break-even or trail modules move them as part of their normal operation.

SECTION 4

Settings reference

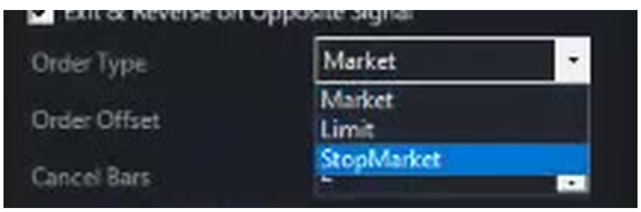
Every setting below appears in the strategy properties dialog. The ones marked **Live** also appear in the settings window and can be changed without disabling the strategy.

4.1 Auto Trade

Setting	Default	Live	What it does
Continuous Mode?	True	Yes	When on, the strategy keeps taking entries. When off, it pauses after each fill and waits for you to re-arm it.
Allow Long Trades	True	Yes	Permits long entries.
Allow Short Trades	True	Yes	Permits short entries.
Exit & Reverse on Opposite Signal	False	Yes	When on, an opposite cross while in a trade flattens the position and enters the other way. When off, the opposite cross is ignored while a position is open.

4.2 Parameters

Setting	Default	Live	What it does
Order Type	Market	Yes	Market, Limit or StopMarket. Limit and StopMarket use Order Offset.
Order Offset	1	Yes	Ticks away from the signal bar's close for Limit and StopMarket entries. Ignored for Market orders.
Cancel Bars	2	Yes	Cancels an unfilled entry order after this many bars. Set to 0 to never cancel.
Quantity	1	Yes	Contracts or shares per entry.
Target	12	Yes	Profit target in **ticks** from the entry price.
Stop Loss	15	Yes	Stop distance in **ticks** from the entry price.



The Order Type selector in the live settings window.

4.3 Break-Even

Setting	Default	Live	What it does
Enable Break-Even	True	Yes	Turns the break-even module on.
Profit Trigger	12	Yes	Once the position is this many ticks in profit, the stop is moved.
Plus Ticks	0	Yes	Where the stop moves to, relative to the entry price. 0 is exact break-even. 2 would lock in two ticks.

4.4 Auto Trail

Setting	Default	Live	What it does
Enable Trail	False	Yes	Turns the trailing stop module on.
Trail Stop	3	Yes	How far behind price the trailing stop sits, in ticks.
Trail Trigger	7	Yes	Profit in ticks required before trailing begins.
Trail Frequency	3	Yes	The stop steps forward once for every this many ticks of new profit. A larger number means fewer order modifications.

Break-even and trail together

Both modules can be on at once. Break-even fires first and moves the stop to entry; the trail then takes over as profit extends. If you find the stop being modified very often, raise **Trail Frequency** rather than widening the stop.

4.5 Session and daily limits

Setting	Default	Live	What it does
Trade 24/7	True	Yes	Ignores the session window entirely. Turn off to use the times below.
Session Start	09:30	Yes	First time of day an entry may be taken.
Session End	16:00	Yes	Last time of day an entry may be taken.
Enable Daily Cut-Off	False	Yes	Turns on the daily profit and loss limits below.
Daily Max Loss	0	Yes	In account currency. When realised plus unrealised P&L for the day falls to minus this figure, the position is closed and no more entries are taken until the next session. Set 0 to disable.
Daily Profit Target	0	Yes	In account currency. As above, but for profit. Set 0 to disable.

Daily limits are measured on this strategy only

The cut-off tracks the cumulative P&L of this strategy instance since it was enabled, including the open position's unrealised P&L. It does not see trades you place manually or from other strategies. It resets at the session close. This is a useful foundation for prop-firm rules but it is not a complete implementation of them — most firms also apply a trailing drawdown that the strategy does not track.

4.6 Moving averages

Setting	Default	What it does
MA Type	EMA	One of twelve: EMA, SMA, WMA, VWMA, HMA, DEMA, TEMA, TMA, T3, KAMA, ZLEMA, LinReg. Both averages use the same type.
Fast MA Length	7	Period of the faster average.
Slow MA Length	21	Period of the slower average.
Show Plot?	True	Draws both averages on the chart.
Fast MA Color	DarkViolet	Plot colour of the fast average.
Slow MA Color	MediumVioletRed	Plot colour of the slow average.

4.7 Platform settings worth knowing

These are standard NinjaTrader strategy properties. The defaults shipped with the Blueprint are listed because several of them matter more than they look.

Property	Default	Why it matters
Calculate	On bar close	Signals are evaluated only on closed bars. Switching to On each tick makes the strategy react intrabar — more fills, more noise, slower backtests.
Bars Required To Trade	20	Raise this if you add an indicator that needs a longer warm-up, or the strategy will throw on the first bars.
Entries Per Direction	1	One position at a time. The bracket assumes this.
Exit On Session Close	True	Flattens 30 seconds before session close. Set false for 24-hour instruments.
Start Behavior	Wait until flat	The strategy will not adopt a position that already exists on the account.
Realtime Error Handling	Stop, cancel, close	On an order error the strategy stops and cleans up rather than continuing blind.

SECTION 5

How it trades

MNQ on 4-tick MeanRenko Heikin Ashi bars. Entries, targets and exits print on the chart.

5.1 The entry

On each closed bar the strategy compares the fast and slow moving averages. A cross above triggers a long, a cross below triggers a short. The entry is submitted using the chosen Order Type, and an unfilled Limit or StopMarket order is cancelled after **Cancel Bars** bars.

An entry is taken only when all of the following are true: AUTO TRADE is on, the direction is permitted by the LONG/SHORT buttons, the clock is inside the session window, the daily cut-off has not fired, and the account is flat.

5.2 The bracket

Once the entry fills, a stop and a target are submitted as real `Order` objects, at **Stop Loss** and **Target** ticks from the fill price. They are tracked through `OnOrderUpdate` and `OnExecutionUpdate`, not through `SetStopLoss` and `SetProfitTarget`.

That distinction is the reason the file is worth its price. Tracking the real order objects is what lets the strategy survive a partial fill, a rejected order, a disconnection or a mid-position reversal without leaving a working order orphaned at the broker while your position is flat — or worse, leaving you in a position with no stop.

5.3 Stop management

While the position is open, two optional modules can move the stop:

- **Break-even** — once profit reaches **Profit Trigger** ticks, the stop moves to the entry price plus **Plus Ticks**.
- **Auto-trail** — once profit reaches **Trail Trigger** ticks, the stop follows price at **Trail Stop** ticks, stepping forward every **Trail Frequency** ticks of new profit.

Both modules modify the live stop order rather than cancelling and resubmitting it, which keeps the broker-side order in sync and avoids a window where the position is unprotected.

5.4 Exit and Reverse

When **Exit & Reverse on Opposite Signal** is on and an opposite cross occurs while you are in a position, the strategy performs a sequence rather than a single action:

1. Cancel every working order for the position.
2. Flatten at market.
3. Wait until the account is genuinely flat **and** all orders have reached a final state.
4. Only then submit the entry in the new direction.

Step three is what most implementations get wrong. Submitting the reversal before the previous orders have settled is how you end up double-sized or with a stray stop working against a fresh position.

SECTION 6

Editing the code: the nine EDIT ZONES

Every place in the file you are meant to change is marked with a banner comment. Search the file for `EDIT ZONE` to move between them. Anything marked `CORE ENGINE` is the order plumbing — read it, borrow from it, but change it last and change it carefully.

Zone	Where	What you do there
0	Top of file	Rename the strategy to your own
1	OnBarUpdate	**Replace the signal. Start here.**
2	CreateMA()	Add another moving-average type
3	State.SetDefaults	Change the default settings
4	State.DataLoaded	Wire up indicators and extra timeframes
5	Input Parameters	Add your own settings to the UI
6	Trail and BE	Change how the stop moves in profit
7	Info Panel	Change what the on-chart card shows
8	Settings Form	Add a field to the live settings window
CORE	System risk management	Order tracking. Do not rewrite lightly.

6.1 Zone 0 — renaming the strategy

To ship this under your own name you must change four things **together**, or NinjaTrader will not compile:

1. The three enums at the top: `NSLTC1_EntryType`, `NSLTC1_MAType`, `NSLTC1_DisplayPosition`.
2. The class name: `public class NSL_TrendCross : Strategy`.
3. The Name property: `Name = "NSL_TrendCross";` in SetDefaults.
4. The file name itself.

The simplest method is Replace All on the string `NSL_TrendCross`, then rename the file, then compile.

> **The mistake everyone makes when cloning the file**
>
> The enums sit **outside** the namespace so the settings form can reach them. If you copy the file to create a second strategy on the same machine and do not rename the enum prefix `NSLTC1_`, you get duplicate-type compile errors that look unrelated to what you changed. Rename the prefix to something unique per strategy.

6.2 Zone 1 — the signal

This is where almost everyone starts. The entire entry logic is two boolean variables, and everything downstream reads them and nothing else:

```
bool IsCrossAbove = CrossAbove(maFastChart, maSlowChart, 1);
bool IsCrossBelow = CrossBelow(maFastChart, maSlowChart, 1);
```

The contract is simple. Set `IsCrossAbove` true when the strategy should want to be long, and `IsCrossBelow` true when it should want to be short. Never set both true on the same bar.

> **Use idx, not 0**
>
> When reading a series inside Zone 1, use the `idx` variable rather than a literal 0. It resolves to 0 on bar close and 1 on tick-by-tick, which keeps your signal consistent in both calculation modes. Using 0 directly will make a strategy that behaves differently in backtest and live.

SECTION 7

Recipes: five changes people make first

7.1 Use RSI reversals instead of an MA cross

Replace the two booleans in Zone 1:

```
bool IsCrossAbove = RSI(14,3)[idx] < 30 && RSI(14,3)[idx+1] >= 30;
bool IsCrossBelow = RSI(14,3)[idx] > 70 && RSI(14,3)[idx+1] <= 70;
```

Nothing else changes. The bracket, break-even, trail, buttons and panel all keep working.

7.2 Add a higher-timeframe trend filter

This is the single most requested addition, because an MA cross with no trend filter whipsaws in ranging markets. Three steps.

First, add the series in Zone 3 (SetDefaults):

```
AddDataSeries(BarsPeriodType.Minute, 15);
```

Second, build the indicator against it in Zone 4 (DataLoaded). Declare `private EMA emaHtf;` near the top of the class first:

```
emaHtf = EMA(Closes[1], 50);
```

Third, gate the signal in Zone 1:

```
bool htfUp = Closes[1][0] > emaHtf[0];
bool IsCrossAbove = CrossAbove(maFastChart, maSlowChart, 1) && htfUp;
bool IsCrossBelow = CrossBelow(maFastChart, maSlowChart, 1) && !htfUp;
```

7.3 Skip dead markets with a volatility floor

Add this to Zone 1 to stop taking entries when the market is not moving enough to reach your target:

```
bool active = ATR(14)[idx] > 4 * TickSize;
bool IsCrossAbove = CrossAbove(maFastChart, maSlowChart, 1) && active;
bool IsCrossBelow = CrossBelow(maFastChart, maSlowChart, 1) && active;
```

7.4 Add your own setting to the properties dialog

Copy this pattern into Zone 5, then give the property a default in Zone 3:

```
[NinjaScriptProperty]
[Range(1, int.MaxValue)]
[Display(Name = "ATR Period", GroupName = "My Filter", Order = 1)]
public int AtrPeriod { get; set; }
```

GroupName creates the section header in the dialog. Any public property with a `[Display]` attribute is automatically optimisable in the Strategy Analyzer — you get that for free.

Serialising a colour needs two properties

If you add a `Brush` property, copy the `MASlowChartBrush` / `MASlowChartBrushSerializable` pair at the end of Zone 5 exactly. Without the second property the colour resets every time NinjaTrader restarts.

7.5 Make the stop adapt to volatility

Zone 6 is the natural home for an ATR-based trail. Replace the fixed tick distance with a multiple of ATR expressed in ticks:

```
int trailTicks = (int)Math.Round(2.0 * ATR(14)[0] / TickSize);
```

Feed `trailTicks` into the existing trail calculation instead of `Stoploss1`. The rest of the module — including the `ChangeOrder` call that keeps the broker in sync — is unchanged.

Create indicators once, never per bar

Always create indicator instances in Zone 4 (DataLoaded) and store them in a field. Creating them inside `OnBarUpdate` is the number one cause of a strategy that runs fine on 500 bars and freezes on 50,000.

SECTION 8

Troubleshooting

Symptom	Likely cause and fix
Duplicate type name compile errors	A second copy of the file, or a clone where the `NSLTC1_` enum prefix was not renamed. See Section 6.1.
Strategy compiles but no buttons appear	The buttons attach on `State.DataLoaded`. Remove the strategy from the chart, re-add it and enable it again. If Chart Trader is hidden, the buttons have nowhere to render.
No trades at all	Check in this order: AUTO TRADE is on; LONG or SHORT is enabled; **Trade 24/7** is on or the clock is inside the session window; the daily cut-off has not fired; the account is flat.
Entry orders keep getting cancelled	**Cancel Bars** is cancelling unfilled Limit or StopMarket orders. Either raise it, set it to 0, or use Market orders.
Stop appears to move constantly	**Trail Frequency** is too low, so the stop steps forward on almost every tick of profit. Raise it.
Strategy throws on the first bars	An added indicator needs a longer warm-up than **Bars Required To Trade**. Raise it above the longest period you use.
Backtest and live behave differently	Usually a literal `0` used instead of `idx` in the signal, or **Calculate** set to On each tick with an order fill resolution that cannot resolve intrabar.
Colour setting resets on restart	The `Brush` property is missing its companion serialisation property. See Section 7.4.
Chart is slow after adding an indicator	The indicator is being created inside `OnBarUpdate` rather than in Zone 4.

> **Where errors appear first**
>
> NinjaTrader writes strategy exceptions to **Control Center** → **New** → **Output**. The strategy wraps `OnBarUpdate` in a try/catch and prints the exception there rather than silently failing, so check the Output window before anything else.

SECTION 9

What it does not do

This list is here so you know exactly what you have before you build on it. Each of these is a reasonable candidate for your free modification.

Gap	Detail
Fixed-tick stops only	Target, stop and trail are all in ticks. An ATR-based version is roughly a twenty-line change in Zone 6 — see Section 7.5.
No higher-timeframe filter	The strategy runs on a single data series. Section 7.2 shows how to add one.
One target, no scale-out	A single bracket per entry. Partial exits at two or three targets require a second order pair inside the core engine.
Fixed quantity	No risk-percent position sizing. Size is whatever you set.
Currency cut-off, not trade count	Daily P&L limits exist. Max trades per day, max consecutive losses and a cooldown after a stop-out do not.
One trading window	A single start and stop time, or 24/7. Multiple sessions and news blackout windows are not built in.
No external alerts	Nothing sends email, Discord or webhook notifications on a fill.
No CSV trade log	Results live in the Strategy Analyzer only. There is no per-trade file export.
No prop-firm trailing drawdown	The daily cut-off is a foundation, not a complete implementation of prop-firm rules. Check your firm's automation policy before using any bot — many restrict or ban fully automated trading.
Button state is not persisted	AUTO TRADE, LONG and SHORT reset to their property defaults when the strategy is re-enabled.

SECTION 10

License, support and your free modification

10.1 What your licence allows

	Personal	Developer
Complete unlocked source	Yes	Yes
Use on your own trading accounts	Yes	Yes
Number of machines	Unlimited	Unlimited
Modify the code however you like	Yes	Yes
Ship it inside client projects	No	Yes
Sell compiled strategies built on it	No	Yes
Redistribute or resell the source itself	No	No
Free custom modifications	One	Two
Free updates for life	Yes	Yes

In plain terms: the file is yours permanently, with no activation, no expiry and no server check. Both licences let you change anything. The only thing neither licence permits is redistributing the source itself — publishing it, sharing it, or reselling it as source. Build whatever you want on top of it.

10.2 Your free modification

Every purchase includes one custom modification, redeemable within 30 days. The Developer Licence includes two.

1. Email your request describing the one change you want.
2. The scope is confirmed in writing before any work starts, so there are no surprises.
3. The modified file comes back to you, typically within three to five business days.

Comfortably inside the free modification: swapping the signal for your indicator, adding a higher-timeframe filter, an ATR-based stop, an extra input or panel row, session or time-window changes.

Quoted separately: multi-instrument or portfolio logic, full custom strategies, scale-out bracket rework, prop-firm drawdown tracking. Owners receive 20% off this work.

10.3 Updates

Any fix or improvement made to the base file is emailed to every owner, free, permanently. Because your copy is source that you control, you decide whether to merge a change or stay on your own version.

10.4 Support

Questions about the code are answered by the developer who wrote it. Support covers installation, compilation, how the code works, and explaining any part of the file. It does not cover trading advice, parameter selection for your instrument, or debugging modifications you have made yourself — though those can be quoted as custom work.

ninjastrategyloader.com