BROKER CONSOLE · MODULE REFERENCE

Pricing and Liquidity

Where a price comes from, what is done to it on the way to the client, and whether it was any good when it arrived. Two menus that are deliberately not one: a quote's *source* and a quote's *treatment* are different decisions, owned by different people, and conflating them is how a desk ends up unable to say which of the two moved.

- Provider registry, symbol map, allocation
- Credentials never read back
- A switch behind a comparison
- A tick recorder with a stated cap
- Three numbers kept three numbers
- Names the half it cannot serve

PRODUCT	MODULE	CONTENTS	VERSION
HonorPro Hub	**Pricing & Liquidity**	**2 menus · 15 tabs**	**1.0 · 24 Sep 2026**

LP Data Feed	Connect Price Feed · LP Switcher · Feed Allocation · Liquidity Bridge · Feed Policy
Feed Manager	Overview · Feeds · Symbols · Price Comparison · Latency · Spikes · Stale Prices · Spread Analysis · Failover · Alerts

01 THE MODULE

Two questions that are never the same question

The Feed Jammer owns what is *done* to a quote on its way to a client — the outlier that is dropped, the spread that is widened, the gap that is handled deliberately. The LP Data Feed owns where that quote *came from* — which provider prices which group and symbol. The screens keep them apart on purpose, and so does the storage behind them.

2
MENUS
The source and its treatment; the quality of what went out

15
TABS
Five on the LP Data Feed, ten on the Feed Manager

4
PROVIDER KINDS
Gateway plugin, feeder, third-party bridge, or the house book

The gateway never touches a tick

Shaping a quote happens in the C++ plugin on the broker's MT5 server, in the microseconds a tick has to spare. This console stores a validated, versioned policy, exports it for the plugin to fetch, and collects the plugin's acknowledgement that it applied a version. That split is the whole safety argument: a Python service that can be restarted and can be slow has no business in the tick path.

So the honest thing a screen can say about a setting enforced elsewhere is two facts, drawn apart: **what is saved**, and **what is in force** — *applied v7* against *saved v8, not confirmed*, or *plugin silent since 09:12*.

A provider model that did not exist before

The older liquidity schema deliberately models **one cover account on the broker's own server**: no list of providers, no symbol map, no per-symbol or per-group routing, no outbound connection. There was an LP concept — a name a plugin was expected to recognise — and no LP administration. The provider registry, the symbol map, the allocation table and the switcher are new state this gateway owns and can answer for.

What it does **not** do is open a session. A manager connection cannot subscribe to somebody else's book, so the session belongs to the server plugin or a feeder beside it — and every screen that would need one says so in the gateway's own words rather than drawing a control that always refuses.

02 MENU · LP DATA FEED

The providers, and which book prices what

Five tabs, answering two questions: who supplies the price, and what this brokerage does to it. Four are the source; the fifth is the policy.

Connect Price Feed `/lp-feed/connect`

Add a provider, map its instruments onto this book, and read what is being priced from it. A provider carries a kind, a status, an endpoint, an account name and a credential reference. The symbol map is one line per instrument — their name, our name — with case preserved, because an MT5 book holds EURUSD.a beside EURUSD; two of theirs onto one of ours is refused.

4 kinds · draft · enabled · disabled · Symbol map · Live reading · Connector checklist

LP Switcher `/lp-feed/switcher`

Move a symbol, or a whole group, from one provider to another. There is no button that switches: choosing a selection builds a comparison — both providers side by side, per instrument — and the switch lives inside it behind a written reason and a typed confirmation. Each switch holds the whole allocation table as it stood before, and Revert restores that copy rather than re-deriving it.

Group or named instruments · Optional from-provider · Reason required · Most recent switch only

Feed Allocation `/lp-feed/allocation`

Which provider prices which group and symbol, as one table — including every row nobody has allocated, which reads *not allocated, so the house book prices it* rather than being absent. Rules take a glob on both sides and the narrower one wins wherever it sits. A save carries the revision it read: a table that moved underneath is a refusal, never a silent overwrite.

One table · Narrower rule wins · Revision-checked save · Unmapped rows counted · CSV export

Liquidity Bridge `/liquidity`

The cover account: one of the broker's own accounts, on his own server, onto which every client's net risk is routed and netted. This screen owns the decision — which flow is covered, inside what limits, at what commission — and the plugin owns the execution: the configuration, the account's status, the cover book by volume, settlement, the journal.

Configuration · Status · Volume · Settlement · Journal

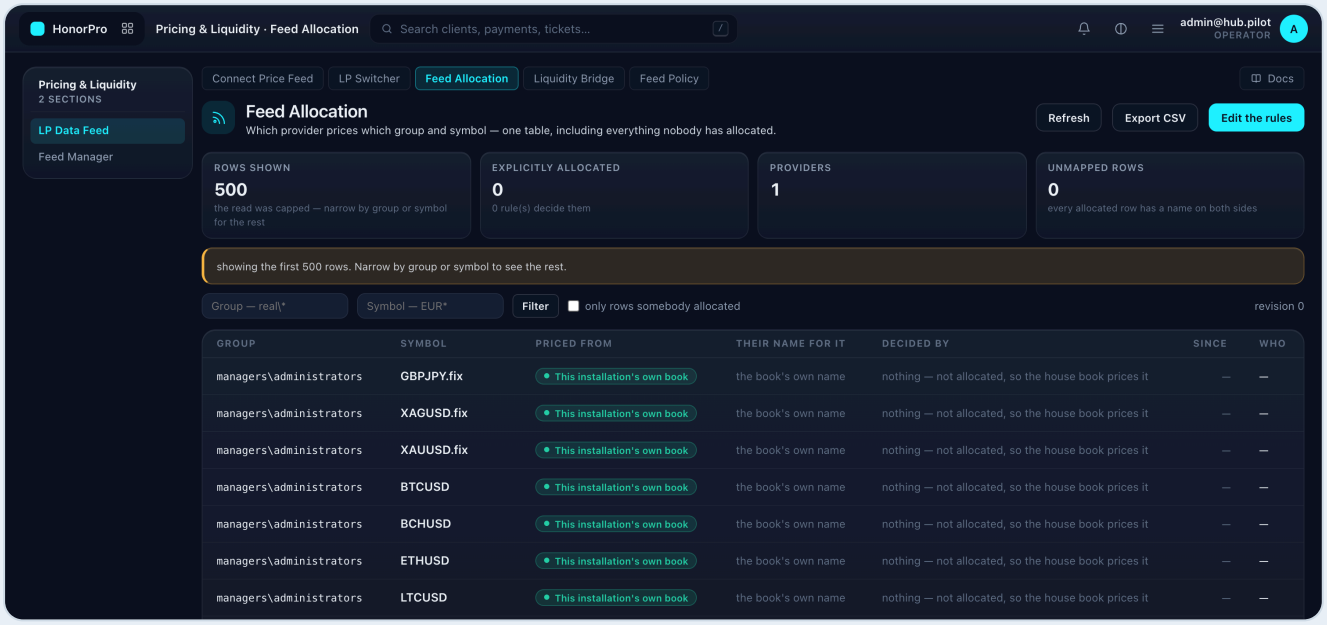
Feed Policy `/feedpolicy`

The Price Feed Jammer's document: a spike filter, gap control, spread engineering, smoothing and a per-symbol failover source, plus session windows, group rules, news windows and per-symbol overrides. Spread is *base* × *multiplier* + *markup, capped, then floored* — the floor last, so a cap saved below it leaves the protective half standing. A simulator shows what the policy would have done to historical bars, and changes nothing.

Spike · gap · spread · smoothing · failover · Sessions · groups · news · Simulate · Plugin status

Configuration, not a connection

The registry is real, answerable state: the provider, its map, its allocation and its switch history. What it is not is a connection — and the screens name, line by line, what a live one would still need and whose job each line is.



Feed Allocation — every group and symbol in one table, with the unallocated rows named rather than omitted.

Whose each missing piece is

MISSING	WHOSE
An endpoint to reach the provider on	The broker
A credential, by reference or stored	The broker
A symbol map, theirs to ours	The broker
A connector on this installation	HonorPro

Switching to a recorded-but-unconnected provider moves the routing; prices arrive when the connector does, and the comparison says so before the switch is confirmed. Depth is a dash on every row for the same kind of reason — this backend port has no market-depth read, and a zero would be an answer.

A credential is never readable back. A provider carries the *name* of an environment variable on the gateway host. Where an installation allows direct entry at all, the password goes straight to a mode-600 file beside the configuration; HTTP answers only whether it is set. A key that looks like a pasted secret is **refused** rather than quietly dropped, because a silently discarded credential is a provider that fails for a reason nobody can see.

The house book

Every installation has one provider it did not create: its own book. It exists so that *which provider prices this symbol* always has an answer, and so a switch away from a real provider has somewhere honest to return to. A provider that still prices something cannot be removed; the allocation table lists what is pointed at it.

04 MENU · FEED MANAGER · 1 OF 2

The product, and the first two measurements

Is the price good: fresh, sane, and the same as everyone else's. The first three tabs are the distribution product — the broker's live quotes sold as a feed, a key per consumer — and the next two begin the quality desk that reads what that feed actually did.

Overview `/price-feed`

Whether the feed is live and on which source, how many symbols are published, how many keys are active out of those ever minted, how many streams are open, how many backend reads the cache has needed and how wide its window is — then the consumer table, with each key's scope, rate limit, call count and last use. A key is issued here and its token is shown once, in that answer and no other.

Live or off · Published symbols · Active keys · Open streams · Backend reads

Feeds `/price-feed?tab=feeds`

The source, the ceiling and how a consumer reads it. **Raw** is the dealer-side quote; **shaped** lays the Jammer's markup over it if that module is enabled and reachable, and falls back to raw rather than go dark if it is not. A ceiling of 1 to 500 symbols per request stops one call reading the whole book and calling it a snapshot. Below it, the delivery figures and copy-ready snippets for a polled read and a live ticker.

raw · shaped · Max symbols per request · Cache window · Key header · 404 · 401 · 403 · 429

Symbols `/price-feed?tab=symbols`

The published allowlist — empty means the whole book — read against every key's own scope, because what a key may read is the allowlist narrowed by the key. The two interesting rows are the mismatches: a published symbol no active key is scoped to, and a key scoped to something not published, which is a standing refusal nobody has noticed.

Allowlist · Keys reading everything · Published, unread · Scoped, not published

Price Comparison `/price-feed?tab=compare`

The raw feed against the shaped feed a consumer on that source receives, per symbol, computed through the Jammer's own spread arithmetic. It is the useful comparison for a dealer — what the client is getting against what the server sees — and it is not a comparison against another price source. What a second one would take is named on the same tab rather than implied.

Symbols compared · Where the two differ · Source in force · What is unavailable, and why

Latency `/price-feed?tab=latency`

What this gateway genuinely controls, over a rolling window of recent serves: the time a server read took, and how old the oldest price in an answer already was when it went out — which is the delay this product adds, and the only figure here that is. Reads started flat while subscribers climb is the shared cache working.

Median · 95th · worst · best · Price age at serve · Cache-hit behaviour · What is not measured, by name

05 MENU · FEED MANAGER · 2 OF 2

What went wrong, and who gets told

Five tabs over the same recorded window. Each of them separates the thing it can measure from the thing it cannot, and the last one turns the other four into rules a person is woken by.

Spikes `/price-feed?tab=spikes`

Ticks that landed further than a threshold you set from the median of the ticks before them — a median, so one outlier cannot move the line it is measured against — plus every crossed quote. A spike is a tick, not a condition: it is raised once, it does not clear, and marking it reviewed records that somebody looked rather than that it did not happen.

Threshold in per cent · Crossed quotes · Per symbol · Review with a note

Stale Prices `/price-feed?tab=stale`

Published symbols that are still being read and whose price has stopped moving, against a threshold an operator sets. Three other populations are kept out of that count and given their own tables: **quiet** symbols nobody is pulling, symbols published but never once asked for, and — when the book is wider than the recorder — the symbols outside the window entirely.

Stale · fresh · quiet · Published, never seen · Tracked against the cap · Threshold presets

Spread Analysis `/price-feed?tab=spread`

Three numbers, kept three numbers. **Observed** is measured from the bid and ask this feed published and is the only measurement on the tab. **Configured** is the spread difference set on the symbol in the MT5 book — a setting the server holds. **Jammer** is the markup policy, also a setting, and one that reaches a consumer only while the published source is shaped. They sit in separate columns because adding them would be a lie.

Now · median · 95th · widest · Symbol-book setting · Markup policy · Shaped would be

Failover `/price-feed?tab=failover`

The three places the publishing path degrades rather than going dark: shaped falling back to raw when the Jammer is off or trips, a warm symbol keeping its last good price through a failed read, and the whole feed answering a clean refusal the moment an operator switches publishing off — without revoking a single key. Each row says whether it is in force today.

Shaped → raw · Cache → last good price · Feed off → refused · No second provider

Alerts `/price-feed?tab=alerts`

Rules over those reads — a symbol stopped ticking, the spread went too wide, a tick moved too far, or the feed stopped altogether — each with its own threshold, symbol scope and severity. They are edge-triggered: a breach raises once and again when it clears, and open incidents are acknowledged. This screen delivers nothing itself; a raised rule goes out on the gateway's own notification path, configured under Server Management.

4 rule kinds · Edge-triggered · Incidents to acknowledge · Periodic sweep, when armed

06 FEED MANAGER · WHAT THE QUALITY TABS READ

A recorder whose cost does not grow with consumers

Until the gateway recorded anything, six of the Feed Manager's tabs were honest gaps: latency, stale prices, spikes, spread, comparison and the rules that would watch them all had nothing to read. The recorder is that read — and the reason it could be added to a hot path at all is where it is called from.

Once per cache fill, never on a hit

The quote path is a loop: an open stream polls the snapshot twice a second, per stream. Recording on every serve would cost subscribers × symbols appends a second — a recorder whose price grows with the audience is exactly what the shared cache exists to avoid.

So a tick is recorded **once per cache fill**, from inside the cache after the backend read returns, and never on a cache hit. At most once per window per symbol, however many consumers are reading. The recorder's cost is pinned to the backend read it is describing, not to the number of subscribers — and the one thing on the serve path is a single counter, which is what makes the cache-hit figure on the Latency tab a measurement rather than a guess.

What it records, and what it refuses to

- **The last tick per symbol** — when the price last *moved*, kept apart from when the symbol was last *read*, because a symbol nobody asked for is not a symbol that stopped ticking
- **A bounded rolling window** per symbol, capped in count and in age, so spikes, spread history and staleness have a real series
- **The timing this gateway controls** — the read time, the age of the cached price when it was served, whether the backend was touched at all

A stated footprint

Three caps, all reported by the recorder itself so a screen can say what it is showing and what it is not: how many symbols are tracked at once, how deep each symbol's ring is, and how old a tick may be before it leaves the window. At the defaults that is a few megabytes, and the cap was chosen against a book that fits inside it.

Over the cap, it **keeps what it already tracks** rather than churning its window or dropping newcomers in silence — and it counts and names the symbols it turned away. Every screen reading it shows that count, so **"no stale symbols" can never quietly mean "the stale one was never tracked"**.

Cold, not empty

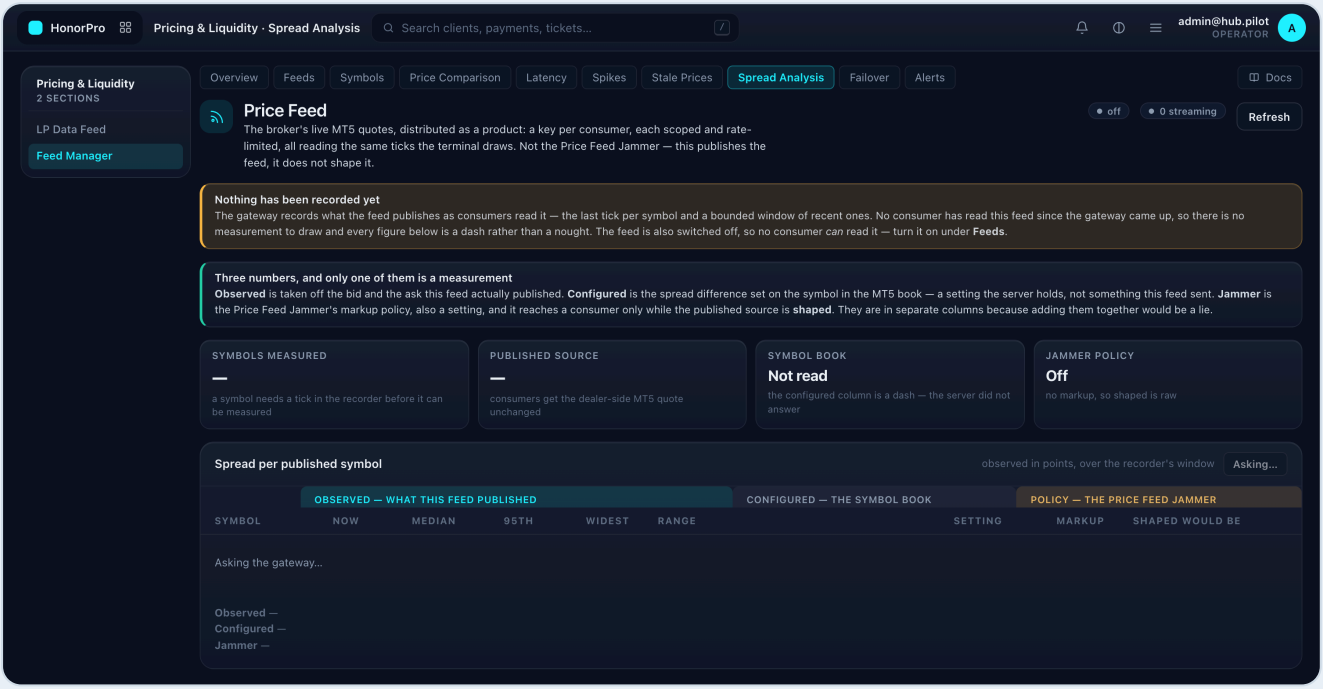
A gateway that has recorded nothing says so once, at the top, rather than six times in six empty tables — and adds that the feed is switched off, when it is, since no consumer *can* read it. Every figure underneath is a dash rather than a nought.

The caps are set on the gateway rather than in the console, and the screens name them rather than hiding them: the Stale tab prints how many symbols are tracked against the cap in force, and the environment setting that widens it. An operator who needs a larger window is told exactly what to change, on the screen where the shortfall shows.

07 FEED MANAGER · THE GAPS IT NAMES

Three things this gateway will not pretend to know

A desk screen is only worth reading if the absences on it are stated. Three tabs carry a real limit rather than a manufactured number, and each says what it would take to close it.



Spread Analysis on an installation no consumer has read yet — three column groups kept apart, and a reason where a nought would fit.

There is one book on this gateway `Price Comparison · Failover`

A cross-provider comparison needs a second book to reach this gateway: the provider registry, its credentials, its symbol map and a quote read per provider — the connector the LP Data Feed menu is waiting for. The same absence is why Failover shows three degradations of the publishing path and states plainly that there is no standby provider to switch to. Across an estate, 360 can put two brokers' published feeds side by side; one gateway on its own cannot.

The window is bounded, so there is no archive `Spikes`

The cache itself holds one price per symbol and overwrites it every window — that is all a distribution cache ever needed. The recorder adds a rolling series, but it is capped in depth and in age: a spike older than the window, or past its symbol's tick depth, has left it. The review mark outlives the tick, which is the honest arrangement — the audit of who looked survives, the raw material for later archaeology does not.

A setting is not a measurement `Spread Analysis`

The symbol book's spread difference is something the server holds; the Jammer's markup is a policy the plugin applies, and only on the shaped source. Neither is a record of what this feed sent. Only the observed column is, and it exists only for symbols a consumer has actually read — a symbol with no series is not a symbol with a narrow spread.

08 HOW IT CONNECTS

Who owns which half

Every screen in this module is one half of a feature whose other half runs on the broker's own server. The console holds the decision and the record; the plugin holds the stream and the book.

TAB	THE CONSOLE OWNS	THE INSTALLATION OWNS
Connect Price Feed, Feed Allocation, LP Switcher	The provider registry, symbol map, allocation rules and switch history	The session to any real provider — a plugin or a feeder, not this gateway
Liquidity Bridge	Which flow is covered, inside what limits, at what commission	Booking the covering position onto the cover account
Feed Policy	The validated document, its version, and the plugin's acknowledgements	Applying the policy to every tick, in the datafeed path
Feed Manager, tabs 1–3	What is published, to whom, and at what rate	The quotes themselves, read once per cache window
Feed Manager, quality tabs	Thresholds, reviews, alert rules and incidents	Nothing — these are pure reads over what the gateway recorded

Licensing

Three separate components sit behind these two menus: the liquidity bridge covers the four source tabs, the feed jammer covers Feed Policy, and the price feed covers the whole Feed Manager. A brokerage that has not bought one does not see its tabs at all — not greyed out, not drawn.

How a change is written

A configuration save is a dry run first, which writes nothing and audits nothing, then a confirmation naming every field that would move — and it carries the revision that was read, so a second dealer editing the same document gets a refusal and a re-read rather than a silent overwrite. Switches and allocations journal who, when and why.

Nothing in the Feed Manager's quality tabs polls, caches or calls the trading server on a hot path. The single backend read in the whole set is the symbol book behind one column of the Spread tab, taken once per request — and when it does not answer, that column is a dash with the server's own reason rather than a borrowed number from the column beside it.