BROKER CONSOLE · MODULE REFERENCE

Custom Platform
under your own name

One branding document, authored once, delivered to the gateway the terminal already reads it from — so the web terminal and the shared mobile app are the same brokerage without being two pieces of work. Beside it: what the terminal offers, where it answers, and the record of every change that went live.

- One document, web and mobile
- Thirteen colour tokens, two palettes
- Draft, diff, publish, restore
- Own-store builds with real artifacts
- Records, never DNS it does not own
- Honest empty states

PRODUCT	MODULE	CONTENTS	VERSION
HonorPro Hub	Custom Platform	2 menus · 14 tabs	1.0 · 24 Sep 2026

HPT White Label Suite	Overview · Branding · Terminal Configuration · Client Features · Markets & Symbols · Trading Connection · Preview · Publishes & Builds · Versions & Activity · Domains · Mobile App · Settings
Live Terminal	Web Terminal — is this installation serving, and where clients find it · Terminal Settings — the defaults it opens with

01 THE MODULE

One terminal, branded once

A brokerage does not get its own build of the terminal. It gets the same build, reading a document that says who it belongs to — and the same is true of the phone. That single decision is what makes this module small enough to operate: there is one place a brand is authored, one place the terminal's behaviour is seeded, and one history of what was published.

2
MENUS
The White Label Suite, and the Live Terminal it produces

14
TABS
Twelve in the suite, two on the live terminal

13
COLOUR TOKENS
The palette, authored twice — a dark set and a light set

Two kinds of field, kept apart

The terminal configuration holds two things that look alike and are not, and the screens label which is which:

- **Seeds** — the value a fresh terminal starts on: theme, lots, slippage, the opening watchlist. A trader who changes one keeps their choice, and moving the seed afterwards does not reach back and overwrite them.
- **Gates** — what the terminal offers at all: the tabs, the allowed order types, whether leverage is shown. A trader cannot turn a gate back on.

Branding is **replaced**, not merged: the console loads the live document, edits it, and sends it back whole. A partial merge would be the surprising behaviour for something authored and delivered as one unit.

Where the documents live

Three public documents on the gateway carry everything a client's software needs before anybody signs in: the branding, the terminal configuration and the client-feature flags. The web terminal reads them on first paint; the shared mobile app asks the same gateway for the same three and caches them on the device.

What the console keeps

Almost nothing. The live surfaces belong to the gateway and are read and written through it; the studio's own store holds only the parts the platform does not keep — the draft being edited, a snapshot of every publish, the activity trail, the domain records and the studio's settings. A fresh installation starts empty and every screen says so rather than inventing a history.

Who may write it

One document brands every terminal on the box, so the write is installation-wide and installation-guarded: a scoped seat — a broker-tenant sharing one live server with another — is refused in one sentence rather than allowed to edit somebody else's brand.

The installation, its brand, and what it offers

Brand it once; web and mobile both take it from here. The first four tabs are the ones a brokerage works through on the day it is set up — what is live, what it is called, how it behaves, and which of the platform's products its clients can see.

Overview `/whitelabel`

The installation at a glance, entirely from live reads: whether the gateway answers and how fast, which backend and build are behind it, the configuration's version and who last set it, how many client surfaces are on, how many publishes there have been, and whether the mobile app code is registered. A setup bar is derived from that real state rather than ticked off by hand, and an unpublished draft is announced here with a door to it.

Gateway reachability · Latency · Setup progress · Last publish · iOS and Android build lines

Branding `/whitelabel/branding`

Six tabs of one document — identity and logos, colours, the login screen, contact and legal, links and social, locales and hosts — with a phone and a browser preview that repaint as you type. Saving is a real, immediate change, so Save & Go Live shows every leaf field that would move, old value to new, before anything is written.

Product and short name · Four image assets · 13 tokens × 2 themes · Login copy · Legal pages · Resolving hosts

Terminal Configuration `/whitelabel/app-config`

The seeds and gates the live terminal reads on boot, edited as a draft: theme, timeframe, default lots, slippage, locale, number format and chart timezone; one-click trading, order confirmation and auto-attached stops; the allowed order types, the leverage display and a decimals override; the four terminal tabs; the broker's links and news sources. Save Draft touches nothing live. Publish shows the diff and puts it through.

Defaults · Trading gates · Terminal tabs · Links & news · Draft · publish · discard

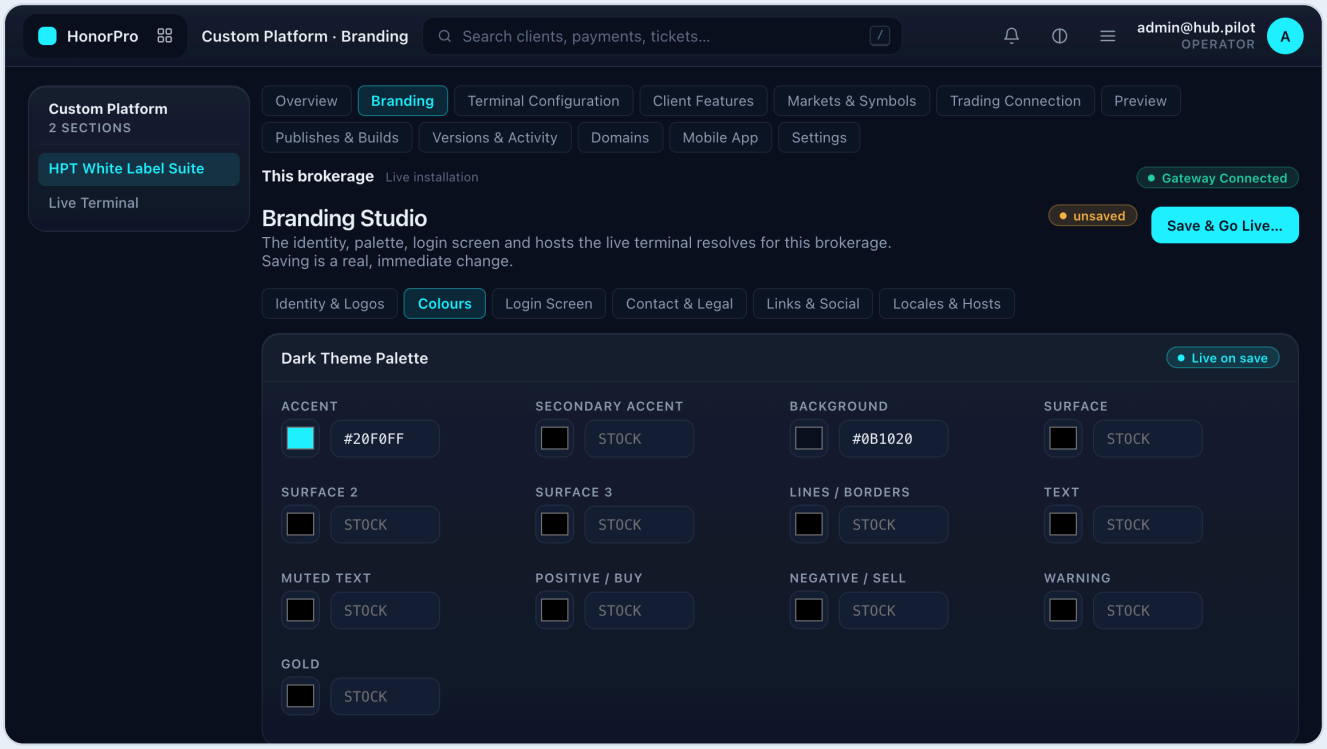
Client Features `/whitelabel/web`

Eight optional surfaces the terminal can offer — PAMM, MAM, Prop Firm, Campaigns, Tournaments, Bonuses, swap-free and the in-terminal cashier — plus the storefront address the Prop Firm tab sends a buyer to. Everything is off until somebody turns it on deliberately, because an empty shelf is worse than no shelf, and applying lists exactly which flags flip first.

8 surfaces · Prop storefront URL · Live on apply · Not a licence, not a permission

What a brand is, as a document

Layout and the screens themselves are the product's and are not brandable. What this document carries is identity, palette, typography, the login screen, the contact and legal footer, the hosts it resolves on, and the few things that are the phone's alone.



Branding Studio, Colours — the thirteen dark-theme tokens; the light set repeats below them.

The sections

- **Identity** — name, short name, tagline, font; both logos, the brand mark and the favicon
- **Palette** — thirteen tokens for dark and again for light; an empty one inherits the stock value
- **Login** — headline, sub-copy, background image
- **Contact and links** — email, phone, entity; four legal pages, four links, six socials
- **Locales and hosts** — the languages offered, the default, and the resolving hostnames

Refusals

An unknown field is refused. A credential-shaped key is refused outright: this file is served through a public door.

The mobile block

One app serves every brokerage, so the mobile-only settings ride in the same document and one save reaches both: a kill switch with its own sentence, whether biometric sign-in is offered, and a tab override that follows the terminal's tabs when left alone.

Only a literal *false* disables the app. An absent value is not a disabled value: a document with no mobile block leaves the app on its defaults — enabled, tabs as the web terminal says, biometric offered. A brokerage cannot take its app down by omission, and the refresh check names a gateway serving no mobile block rather than calling it off.

The book, the pipe, the picture and the history

Four tabs that answer four different questions about the same installation: which instruments exist, whether the connection behind them is healthy, what a client will actually see, and what has been shipped.

Markets & Symbols `/whitelabel/markets`

The trading server's real instrument list — name, description, category and digits — shown read-only, because the server decides which symbols exist and how they are grouped. The one symbol-shaped thing the terminal configuration does control is the default watchlist, ordered here and published through the same diff as any other change. Categories are counted, not edited.

Symbols · Watchlist · Categories · Read-only book · Drafted watchlist

Trading Connection `/whitelabel/trading`

The gateway this console and the live terminal both stand on, probed for real from this machine: the gateway address and operator account, the backend and build behind it, a latency measured around the health call rather than asserted, a row per check with its own detail, and the capabilities the gateway reports from its own probe of the trading server. The connection settings come from the box's environment and are shown, not edited — there is no second place to configure them.

Test Connection · Per-check detail · Measured latency · Server capabilities

Preview `/whitelabel/preview`

The real configuration under the real branding, on iPhone, Android, tablet or desktop, on the Home, Watchlist or Trade screen, in the theme the configuration seeds or either one forced. The honest control is the last one: **Live** is what a terminal boots into today, **Draft** is what it would boot into after the pending publish. Market figures in the picture are placeholders and the caption says so.

4 devices · 3 screens · Live or draft · Theme override

Publishes & Builds `/whitelabel/builds`

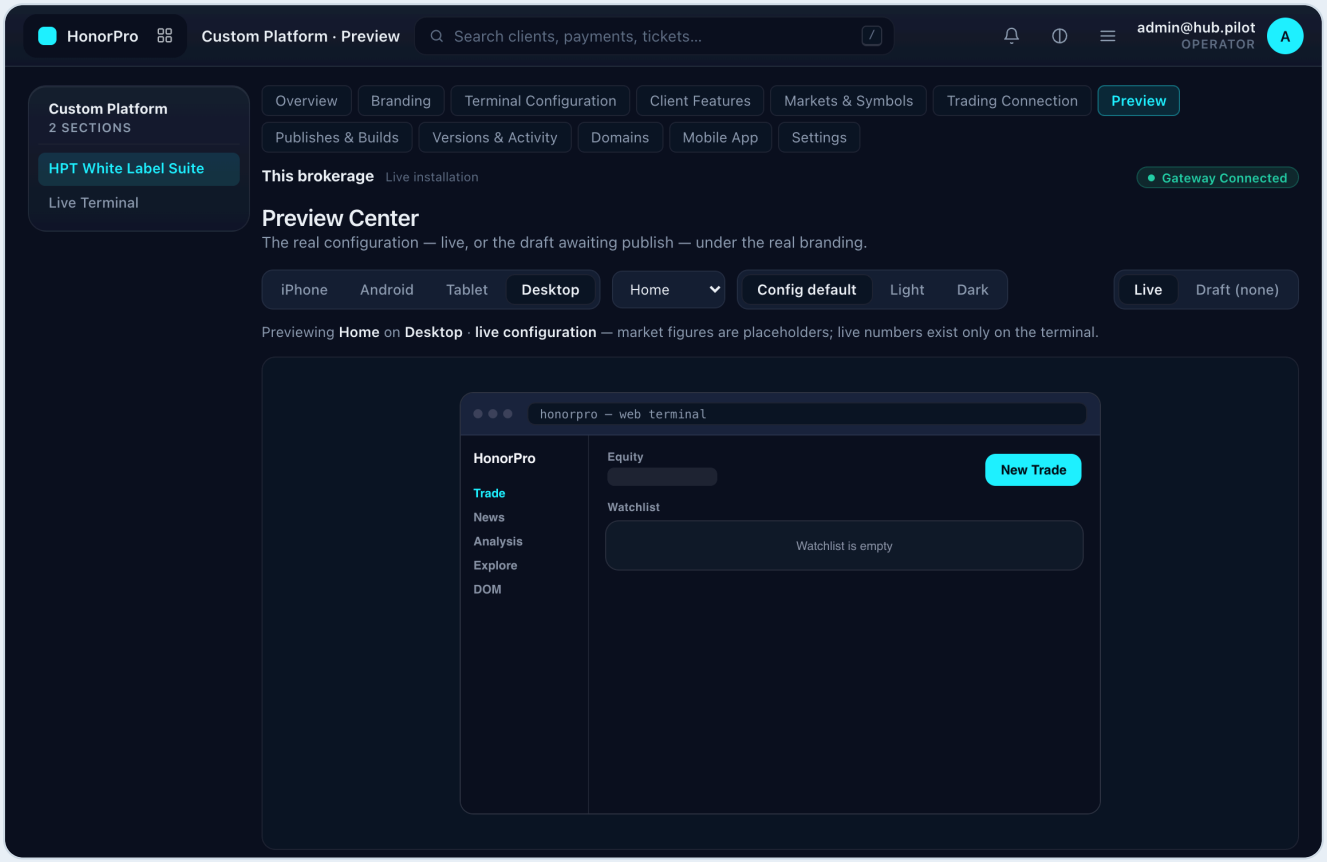
Two real histories side by side. Every web row is a configuration write the gateway accepted, with its version, whether it was a publish or a restore, how many fields moved, and by whom. Every mobile row is a job a build runner claimed — platform, target, version and build number, state, runner, and the artifact it uploaded, downloadable here. A debug-signed build is labelled as one.

Web publishes · Own-store builds · Artifact download · Live job polling

05 MENU · HPT WHITE LABEL SUITE · PREVIEW

What a client will actually see

Twelve screens of settings come down to one question a broker asks on the day they buy. Preview answers it from the same two documents the terminal reads — the branding and the configuration — on the device and the screen of your choosing.



Preview Center on the pilot — the brokerage's own name and accent over the live configuration. The watchlist is empty because this installation seeds none, and Draft reads "(none)" because nothing is pending.

What the controls do

Four devices, three screens, and a theme that is either the configuration's own seed or one you force for the look of it. The last control is the one that matters: **Live** is what a terminal boots into today, **Draft** is what it would boot into after the pending publish — and when nothing is drafted the switch says so and refuses to be pressed, rather than showing a draft that does not exist.

What it is not

It is not the running terminal. The frame is drawn from the two documents, so it answers what the broker configured and not what the market is doing: the screen's own line says market figures are placeholders and live numbers exist only on the terminal. An empty panel here is an empty setting, not a rendering that failed.

The record, the addresses, the phone and the desk

The last four tabs are about accountability and reach: what was published and by whom, where this installation answers, how a client gets the app, and who to ask before any of it changes.

Versions & Activity `/whitelabel/versions`

Every publish this console made, as a card carrying its changes, its moment and its author, with the full list behind View Changes. Restore compares that snapshot against what is live *now*, shows that diff, and on confirmation writes it back — which is itself a publish, recorded as a new version. Under it, the activity log: real actions only — publishes, feature flips, records, connection tests.

Snapshots · Diff on restore · Activity trail · Nothing that did not happen

Domains `/whitelabel/domains`

Your team's record of which hostnames point at this installation, with a purpose, a note, and a status in the operator's own vocabulary — in use, reserved, retired. The Hub manages no DNS and issues no certificates: nothing here probes or provisions, and removing a record changes nothing at your provider. The hosts that actually resolve to this brand are the ones on the branding document.

Operator-maintained · Purpose and status · Who recorded it, when

Mobile App `/whitelabel/mobile`

Two halves. The **shared app** shows what the phone pulls and when, a refresh check that fetches the three public documents exactly as a client's phone would and reports what came back against what the console holds, the app code and links clients enter it by, and the mobile-only settings. The **own store app** is for a brokerage that wants its own listing: store identity, signing material, builds, submissions and runners.

What the app shows · Client access · Mobile settings · iOS · Android · Builds · Submission · Runners

Settings `/whitelabel/settings`

The studio's own console-side policy, and labelled as such: who answers for the terminal's look, whether a publish or a silent gateway raises a notification through the Hub's configured channels, and a contact list of who to ask before a publish. It grants no console access — console users are managed under Team.

Studio owner · Publish notifications · Gateway-offline notifications · Contacts

07 MENU · HPT WHITE LABEL SUITE · GOING LIVE

Nothing goes live without being shown first

Every write in this module reaches real clients the moment it lands, so every write is preceded by the same three steps: a draft that costs nothing, a diff against what is live, and a record of what went through.

The path a change takes

- **Draft** — the terminal configuration is edited console-side and saved without touching the terminal. The Overview says a draft is pending and names how many fields it holds.
- **Diff** — publishing opens the list of changes the gateway would receive. Branding and client features show the same list before they are applied, field by field and flag by flag.
- **Publish** — the document is written through the gateway, and the console snapshots what it accepted.
- **Restore** — an earlier snapshot can be put back, diffed against what is live now, and is recorded as a new version rather than rewriting the old one.

A publish is refused nowhere and announced everywhere: a gateway that cannot be read leaves the configuration screen with **nothing safe to edit** and says so, rather than offering a form over a document it does not have.

Own-store builds

The Hub cannot build a mobile app itself — iOS needs a Mac with Xcode, Android its own toolchain, and neither belongs on the machine that runs a control plane. So the Hub keeps the store identity, the signing material and the store credentials, all encrypted at rest and never returned by any read, and renders the build manifest. A runner enrolled with a one-time token claims the job, builds, streams its log back and uploads the artifact, which is stored with the hash the Hub computed on receipt.

What it will not pretend

A platform with no runner online says so. A job is queued until somebody takes it, and nothing simulates progress. Submission is real or honestly absent: Android goes to a Play track through the store's own API when a service account is set, iOS uploads from the macOS runner with Apple's tools, and with no credentials the screen says *not configured* — never *submitted*.

08 MENU · LIVE TERMINAL

What is actually being served

The suite next door decides what the terminal should be. This menu answers whether it is: whether this installation is serving anybody, where clients find it, and the defaults it opens them on.

Web Terminal `/web-terminal`

Two answers and one door. **Are we serving?** — the gateway decides that, and when the answer is no it lists which settings are still missing, so a freshly installed box is walked through the rest of its setup by the thing that is failing. **Where is it?** — an address on the brokerage's own domain, kept in this component's configuration, which the Agent already carries to the machine serving it.

The address is probed from the Hub rather than from the operator's browser, deliberately: a page cannot tell a dead host from a cross-origin refusal, and the question is whether *clients* can reach it. Beside it, the seven client surfaces — five tabs across the top and two sections of the account zone — each with a door to the desk that owns its switch, and the brand shown rather than edited.

Serving · demo · blocked · silent | Still-missing list | Reachability from the Hub | Build and channel | Licensed concurrent sessions

Terminal Settings `/terminal-config`

The same document the White Label Suite drafts and publishes, reached directly for a desk that wants to edit it in place: theme and locale, the ticket's prefilled lots and slippage, number format and chart timezone, the four tabs, the allowed order types and price decimals, the starting watchlist, the news sources and the help-menu links. It is gateway state, not console state — loaded, edited and written back whole — and open terminals pick a change up on their next load.

Defaults | Tabs | Trading | Watchlist | News sources | Help links

Neither tab is licence-gated in the way the products are: the terminal is part of the installation's own desk, and what the gateway enforces is the operator's **terminal** scope. A session without it is refused, and the screen says which scope it wanted.

Four answers, not two

Serving clients, serving simulated data, not serving, or the gateway is silent — four states rather than a green light and a red one, because a demo box and a broken one are not the same problem. The reasons and the still-missing list are drawn only while something is actually wrong, so a healthy installation is a quiet screen.

Where each switch lives

The seven client surfaces are read-only here and each carries a door to the desk that owns it, so the operator who turns PAMM off is looking at the PAMM pools while they do it. A surface that is off is hidden from traders and nothing else: the desk behind it keeps running, and its licence is a separate question. A concurrent-session ceiling is shown only when the licence states one — a limit nobody was sold is not a limit to draw.

09 HOW IT CONNECTS

One document, three readers

The console writes through the gateway; the gateway serves the terminal, the phone and anything else that boots under this brand. Nothing in this module is mirrored console-side except the parts the platform does not keep.

TAB	READS	WRITES
Overview, Trading Connection	The gateway's health, configuration and feature documents	Nothing; a connection test is recorded on the trail
Branding	The live branding document	The whole document, back to the same file the terminal reads
Terminal Configuration, Markets & Symbols	The gateway's terminal configuration and its read-only symbol book	A console-side draft, then the published document
Client Features	The client-features document	Only the flags that changed
Versions & Activity, Domains, Settings	The studio's own store	Snapshots, records, settings — never the trading server
Mobile App	The three public documents, as a phone sees them, plus the broker directory	The mobile block of the branding document; build and submission jobs

Where the branding file lives

Beside the licence, unless the deployment names another location. The write computes that path exactly as the public read does, so the two can never point at different files, and it is written whole with a temp-and-replace so a crash cannot leave a half-branded terminal.

The app code

A short code a client types on the app's first screen resolves against HonorPro 360's broker directory to find this gateway. The code and the public gateway address are reported there, and what 360 holds is the real one — a value an operator set centrally wins, and a code another brokerage already holds is theirs. Until it is registered, the links and the QR do not exist and the screen says so.

Fails closed, not loudly: the terminal reads these documents before anybody has signed in, so an unreadable one resolves to the fresh-installation answer rather than breaking a booting terminal. That is why the console, not the terminal, is where a missing document has to be visible — and why the Overview surfaces every read that failed instead of averaging them into a status light.