OEM White-Label Program: What to Know Before You Sign

What an OEM needs to know before signing a white-label agreement: what can be branded as yours, what stays fixed, and how much work is involved.
Audience: prospective OEM and white-label partners.

What the Program Is

A white-label OEM agreement lets you ship the FrameworX platform as a product of your own. Your end users install your product, open your engineering IDE, see your name and artwork, follow your help links, and accept your license agreement.

The platform underneath is unchanged: you inherit the full product, its connector library, its AI tooling, and its release stream. You are not maintaining a copy of the software in parallel, there is no separate codebase and no branch of your own to keep in sync with ours. Your branding is applied to the same product we build, so every release arrives already carrying your name, your artwork, and your license.

What You Can Make Yours

Area	What you control
Product name	Your name replaces ours across the product, the Windows shortcuts, the install folder and the uninstaller
Icons and artwork	Your icon set and imagery, in light-theme and dark-theme variants
Themes	Which visual themes ship, and which is the default after install
Solution Manager	Your artwork and layout, reused on the runtime startup dialog
Product editions	Your edition names and icons, mapped to the FrameworX families
Help and forum links	Every in-product help link redirected to your pages
Startup splash	Your splash window at runtime startup
Installer	Your branded setup, your publisher identity, your install path
Digital signature	Signed with your certificate, so the product is yours to Windows
EULA	Your end-user license agreement, shown during installation
The AI tools and the AI Tutor	Every AI tool that ships with the platform is rebranded to your product and your company: AI Tutor, AI Designer, AI Runtime and the rest of the AI toolset. This includes the names your users see when they connect those tools in Claude Desktop, and the local tier recognition files the AI Tutor writes on the learner's machine

Nothing on this list is held back to protect our brand. The rule is consistency, not preservation: a name has to change everywhere it appears at once, so you choose the names and we apply them across the whole build.

What Stays Tatsoft

A defined set of secondary module icons and internal module names carries no customization path. These are not the surfaces your users meet first, but they are fixed, and it is better to see the list during evaluation than to discover it in validation. We will show it to you.

The platform engineering, the release stream, and the reference technical documentation also remain ours.

What You Provide

Brand assets: your product name, icon set, Designer and Solution Manager imagery, splash image, theme choice, and your edition names mapped to ours.

Two legal-commercial artifacts: a code-signing certificate held in a cloud HSM (Hardware Security Module, the certified hardware a signing key has to live on; Azure Key Vault is the common example), and your EULA.

Two pieces of standing infrastructure. These are the ones most often underestimated:

- **A documentation host of your own.** In-product help links and the AI assistant both send users somewhere for depth. Without your own host, your users land on Tatsoft pages mid-session and your brand breaks in front of them.
- **A first-line support tier.** Our support serves you with contractual priority, but the agreement assumes you filter basic end-customer questions before escalating.

What We Deliver

You send us the brand assets and the names; we do the rebranding work. We apply your branding to the product build (the product surface, and the AI layer when your program includes it), produce the branded installer, review your EULA, and validate the result before it ships.

You receive a product that installs and runs as yours, plus the setup project itself if you want to own your own packaging. The build is parameterized for rebranding: your name, icons, and assets are configuration, not a code change. Producing your own signed setup is therefore a packaging step, not a second product for your team to develop and maintain.

Your end customers still need Windows, administrator rights to install, and the .NET runtime the platform is built on. Your installer can bundle that dependency so they never see it.

The Decisions to Close Before Contract

Four questions move cost and timeline more than anything else.

1. **Whose certificate signs the product, and who holds it?** Certificate authorities no longer issue code-signing certificates as a file you can download: the key has to live in a cloud HSM, and the signing happens there. Either you sign the files we send you, or we need access to your signing certificate, and if you want us to build your installer, that access is required. This has cost, lead time, and a security review on your side. It is the single worst thing to leave until after signature.
2. **Who owns the packaging, you or us?** We can produce your installer, or hand over the setup project so your team builds it. This decides who controls your release cadence, and it is coupled to question 1.
3. **Do you stand up your own documentation and first-line support?** Both are assumed by the agreement. If the answer is not yet, that is a real project on your side and it should be scoped now, not discovered later.
4. **How far does your brand extend?** The product surface is the minimum. Extending it to the AI layer (AI Tutor, AI Designer, AI Runtime and the rest of the AI toolset) means every one of those tools reaches your users under your product name. It is a larger piece of work, and we carry it out from the names and assets you provide, so raise it early: it adds scope and lead time to the first build.

Where the Detail Lives

Behind every line above there is a precise engineering specification: exact image dimensions, the build and packaging contract, the rebranding mechanics, and the validation checklist a build passes before it ships. We work from that material to produce your build. The parts your own team needs (packaging, if you choose to own it) are released to your technical contacts once the agreement is in place.

Next step: tell us which of the four decisions are already settled on your side, and we will scope the rest with you.

In this section...
