EVC Checkout

User Manual

Your products live in Invoice Ninja . EVC Checkout displays them on WordPress with images, descriptions, prices, and buy buttons, and takes payment through an on-site Stripe checkout. The customer never leaves your domain.

Version 4.6.7

What it does

- Products live in Invoice Ninja; EVC Checkout renders them on WordPress with images, descriptions, prices, and buy buttons.
- Stripe-first checkout: Buy goes to an on-site checkout page with an embedded card form (PaymentIntent, confirmCardPayment).
- Invoice Ninja only receives the client and a PAID invoice after money actually changed hands.
- Affiliate refs (?r=CODE) ride through the buy flow into the payment metadata.

Requirements and installation

- WordPress 5.8 or newer, and PHP 8.2 or newer.

Both of those are written in the plugin header, which is the line WordPress itself reads: the Plugins screen warns you before activation on a server that is too old, and so does the update check. If you go ahead anyway, or if the host changes PHP underneath you, EVC Checkout does not half-load and it

does not white-screen. It stops, and it prints one notice naming the version your site is running, the version it needs, and the fact that nothing was changed and no setting was lost.

On an older PHP the fix is one line to your host: move this site to PHP 8.2 or newer. Most hosts do it from a dropdown in their control panel.

- Invoice Ninja (v5 or hosted) with an API token
- A Stripe account (secret + publishable keys)

Upload the zip you downloaded through Plugins → Add New → Upload: evc-checkout.zip from the free download, or evc-checkout-pro.zip from the paid order. Or unzip it into wp-content/plugins/ so the plugin lives at wp-content/plugins/evc-checkout/evc-checkout.php. Then activate EVC Checkout.

Free edition and paid edition

The plugin comes in two packages, built from the same source. Both are plain readable PHP. Nothing is encoded, nothing expires, and there is no extension to ask your host for, so nothing in either copy waits on a setting you do not control.

The free edition is a working checkout. The paid edition adds the modules sold separately: Registry Keys, which hands a buyer a software key in the receipt, and Abandoned Carts, which sees the checkouts that did not finish and sends one friendly nudge. In the free edition those two screens are still there, drawn in grey with their real columns and four sample rows, so you can see exactly what they do before you decide.

The key comes in the receipt email and is pasted once on EVC → Settings → Licence. It is checked inside the plugin, from the key itself, with no call to any server, so a shop with no internet still works. A key pasted into the free edition is accepted and told the truth: the modules are in the paid package, and installing that package over this copy keeps every setting.

Getting your credentials (Stripe, Invoice Ninja, Groq)

EVC Checkout runs entirely on your own accounts. Three services, one optional. Everything below is free to start, Stripe and Groq only charge/limit on live volume.

- Invoice Ninja (product catalog + invoicing): self-host v5 or use the hosted plan at invoiceninja.com. In Ninja go to Settings → Account Management → API Tokens and generate a token, that is the API Token you paste into EVC → Settings. Your Ninja URL is where you log in (e.g. https://billing.yourdomain.com).
- Stripe (card payments): dashboard.stripe.com/apikeys gives you the Secret key and Publishable key, paste both into EVC → Settings. For the webhook, go to dashboard.stripe.com/webhooks → Add endpoint,

enter https://YOUR-SITE/wp-json/evc/v1/stripe-webhook, select the checkout.session.completed and payment_intent.succeeded events, then copy the signing secret (whsec_...) into Settings → Stripe Webhook Secret. This is what marks invoices PAID after real money.

- Groq (optional, AI Recovery Emails): create a free key at console.groq.com/keys and paste it into Settings → AI Recovery Emails. The free tier is enough for typical abandoned-cart volume. No key or a failed call simply falls back to the fixed reminder template, so this is always safe to enable.

Press Test the assistant once. Under the key in Settings is that button. It asks one question the way a buyer would and shows either the answer or the exact reason it failed, in words: no key saved, key refused by Groq, Groq busy, this site unable to reach Groq (ask the host about outgoing connections), or a retired model name. Do that when you set the key up, not after a buyer tells you the box is not answering. Save your settings first, so the button tests the key that is stored.

AI model names expire. EVC Checkout tries three names in turn before it fails, so a retired one at the front of the list costs one wasted request rather than a dead feature, and the evc_assistant_models filter lets you put your own names first.

In WordPress: EVC → Settings holds the Invoice Ninja URL, API token, company key, currency, Stripe keys and webhook secret, and the Groq key. EVC → Products lists your Ninja catalog (hide items, flag digital products as Key). EVC → Overview shows the full shortcode reference plus this manual.

Licensing: Free vs Unlimited

- Free: the full core checkout, on-site Stripe payments, Invoice Ninja catalog, product shortcodes, checkout themes, affiliate ref forwarding (?r=CODE). Up to 5 products; one site.
- Unlimited: everything in Free plus Abandoned Carts, AI Recovery Emails, Registry Keys and Custom CSS, with unlimited products. Unlimited sites, lifetime updates, priority email support. The current offer is on the product page at evcapps.ca.
- How the key works: paste it into Settings → EVC License Key. Verification is offline (checksum, no phone-home) and applies instantly. Unlicensed installs simply keep the core checkout; pro admin pages show an upgrade notice.
- Two packages, one key. The free package does not contain the paid modules at all: Registry Keys, Abandoned Carts and the AI recovery

emails are not in those files, so nothing can be switched on by editing them. The paid package is the same plugin with those modules in it. Your key is read and checked in both, so if you buy after installing the free copy, the key is accepted there straight away and the paid package switches the modules on with that same key. The free copy carries the download link itself, on EVC → Settings → Licence. A paid screen in the free copy says exactly that rather than showing an empty page.

Overview (EVC → Overview)

- One dashboard: build version, products on site (visible vs. total), full shortcode reference, manual PDF + Markdown (shipped with the plugin under docs/).

Settings (EVC → Settings)

```
Invoice Ninja URL         Base URL of your Ninja install
API Token                 Ninja API token (Settings → API Tokens)
Company ID                Ninja company id (usually 1)
Currency                 Lowercase ISO code, e.g. usd
From Email                Address used on outgoing plugin email
Stripe Secret Key         sk_live_... / sk_test_...
Stripe Publishable Key    pk_live_... / pk_test_...
Checkout Page URL         Page hosting [evc_checkout], e.g. /checkout/
Stripe Webhook Secret     whsec_... (see Webhook below)
Buyer Confirmation Box    your own statement, ticked before paying;
                          a host setting, a minimum version, any
                          wording of yours. No statement = no box

Appearance is not here. One row, Checkout Appearance, links to the
Design tab, which owns the theme, colours, corners, spacing, type,
button shape and custom CSS.
```

The tab row

Every EVC screen carries one row of tabs under the logo: Overview, Products, Invoices, Send Invoice, Abandoned Carts, Registry Keys, Design, Insert, Settings. Click a tab to switch screen.

Each tab is a normal link that loads that screen, so the Settings form and the Products table keep their own submissions and behave exactly as before. The sidebar keeps its own list too; either route works.

Design (EVC → Design)

A point-and-click design panel, so nobody has to write CSS to make the checkout look like it belongs on the site. Five presets, six colours, five sliders, five dropdowns, a live preview, and no Save button to

remember.

Since 4.3.53 this screen owns everything about how the checkout looks. The theme, accent colour, card background and Custom CSS fields that used to sit in Settings have moved here, so there is one place to look. Settings now just links here.

Since 4.3.66 every control saves as you move it. There is no button to press and no page reload: move a slider, and the change is stored a third of a second later, with the time it saved shown above the preview. The Save design button is still there for anyone who likes pressing one, but nothing depends on it.

Presets

- Clean: white cards, brand blue accent. The default.
- Soft: cream cards, rounder corners, a gentle shadow.
- Bold: dark cards, tight corners, heavy headings.
- Minimal: no borders, square corners, outline buttons.
- Night: deep navy, teal accent.
- Clicking a preset fills every control below it; the controls then let you bend that look in any direction.

Typing an exact colour

A brand is usually specified as a code, so every colour row has a box beside the swatch you can type or paste one into: #000000, or 000000 without the hash, or the short form #000. A code that is not right for a colour, five digits being the usual slip, is refused with six digits, like #000000 rather than quietly turned into something else. The swatch and the box always agree, whichever one you touched last.

Controls

Base	Light or Dark, the starting palette
Accent	buttons, links, checkboxes, focus rings
Card background	product cards and the checkout form panel
Input background	text fields inside the form
Borders	card and field outlines
Text / Secondary	headings and body copy, then descriptions
Card corners	0 to 32px
Input corners	0 to 24px
Card padding	12 to 40px
Space between cards	8 to 32px
Product image height	120 to 320px, cropped to fill
Type	theme font, Acme, System UI, Serif, Monospace
Heading weight	regular through heavy
Card shadow	none, soft, strong
Button style	solid or outline
Button width	fit the label, or full width
Custom CSS	anything the controls do not cover

How it applies

Your choices are saved as tokens and printed as a single style block, about 700 bytes, in the page head: CSS variables on the checkout plus a handful of scoped rules. No images, no framework, no extra file request, nothing to compile.

The tokens are declared on :root and inherit, so they reach every EVC surface on the page, including the buy pop-up and the success panel. The checkout markup reads the tokens itself now instead of carrying its own numbers, so corners, padding, spacing and type stay in step.

The scoped rules win over the theme's own inline values, which is why a slider can change corners, spacing, type and button shape without the page ever rearranging itself. Colour, corners and type only: the layout of the checkout is not yours to break from this screen.

Custom CSS

The escape hatch, below the controls it complements. It is loaded after the design styles, so it wins, and it is applied to the preview as you type, so you can see whether it does what you meant before trusting it on the shop. Wrap it in .evc-checkout { ... } to keep it away from the rest of the site, and set any --evc-* variable from there.

Custom CSS is part of the Unlimited license. On a free install it is typed and stored but never printed, so nothing is lost: enter a key and it starts applying. The screen says exactly that.

Reset, preview, and existing sites

- Reset to preset discards your tweaks and restores the selected preset exactly.
- The live preview is now the thing itself, not a drawing of it. It runs the same style block your shop prints, on the real card and billing markup, with two products so the space between them is visible. Before 4.3.66 it was a hand-painted copy, which is why a change could look right here and wrong on the site.
- Undo takes back the last change, twenty deep. Reset colours, Reset shape and Reset type put one group back to the selected preset without touching the rest, so a colour that went wrong does not cost you your corners and spacing as well.
- Base is Light or Dark, and it does what it says: it swaps the five surface colours for that base and leaves your accent alone. Before 4.3.66 it changed a stored word and nothing on screen.
- A footer note is still true: your theme's own CSS can reach into the checkout, so glance at the real page once after a big change.
- An existing site keeps its exact palette until the panel is saved: the legacy theme, accent and card-background options still decide, including the original dark palette. Saving writes those three options too, so nothing else reads a stale colour. They are options now, not fields in Settings.

Shortcode reference

Create a page and drop in one of these. The same table is on the settings screen.

```
[evc_products]                    All visible products, stacked sections
                                  (grid: columns="2" or columns="3")
```

```
[evc_product id="123"]               A single product card
[evc_buy_button id="123"]           Buy button for one product
[evc_buy_button product="Custom"
                price="35.00"]      Ad-hoc item buy button (legacy)
[evc_invoice_history                Customer invoice history
                email="a@b.com"]    with PDF download links
[evc_field id="123" field="price"]  One field of one product, anywhere
[evc_products group="category"]     One section per category
[evc_products                       Only one category
            category="Personas"]
[evc_checkout]                      The checkout page itself
```

Legacy tags from earlier builds, [tim_products], [tim_product], [tim_buy_button], [tim_checkout], [tim_invoice_history], [tim_field], still work as aliases. New pages should use the evc_ tags.

Product categories

Invoice Ninja has no product category field, so EVC Checkout reads the category from one of the product's four Custom Values, the same mechanism the Persona field already uses.

1. In Ninja open Settings › Custom Fields and rename one product custom value to Category.
2. On each product type the category there, Plugins, Personas, Bundles, whatever you sell in.
3. In WordPress open EVC → Settings → Product Categories and pick that same slot (Custom Value 1–4).

```
[evc_products group="category"]      one section per category
[evc_products category="Personas"]   only the personas
```

Grouping is alphabetical and anything without a category lands at the end under Other, so a half-filled catalogue never opens with a blank heading.

The slot you pick is excluded from persona detection. Without that, a product carrying Plugins as its category and WEB as its persona would have been served to the site as persona Plugins.

Nothing is required: leave Product Categories on No categories and every shortcode behaves exactly as it did before.

[evc_field] in detail

Output ONE field of ONE Ninja product in any page, widget, template part, or page builder block, for custom layouts that don't want a card.

```
[evc_field id="NINJA_ID" field="price"]
[evc_field name="Product Name" field="price" format="raw"]
```

- Fields: item (or name), description (or notes), price, image, tags, custom1-4, id
- price takes format="raw" for the bare number (default is styled, e.g. USD 49.00)
- Lookup by Ninja id or exact product name

The credit line

At the very foot of the checkout page there is one small muted line: Powered by EVC Checkout, linked to the plugin page through the download tracker so clicks can be counted. It sits under the buyer's own shop, never above it, and it is twelve pixels of grey, not a banner.

This is the free version's line. Put a licence key in and it does not appear at all, because a site that has paid for the plugin should not be carrying its advert. Untick Powered by EVC Checkout in Settings and it is gone on the free version too. A developer who wants a different address can filter it:

```
add_filter('evc_credit_url', fn() => 'https://your-site.ca/');
```

Nothing else about the plugin phones home: no usage count, no visitor list, no version report.

The sales assistant in the cart

Under the total on the checkout page, above the card fields, there is a small box that says Ask about this order. A buyer can type a question there, or press one of three starting questions, and get two sentences back before they decide. It is the last place a sale is lost and the last place most stores put anyone.

What it is allowed to know

Your own facts, and nothing else: the store name, the currency, the item on the order, its price, the description you wrote for it, the policies this store publishes, and your support address. All of that is rebuilt on the server for every question, so nothing a buyer sends can change what it knows.

Since 4.4.7 the questions that can be answered from your own facts are answered without asking anybody: a price question gets the price on the page, character for character, a what do I get question gets your own description, and a refund, licence or delivery question gets the exact sentence from the policy you publish, in quotation marks. Those answers are instant, cost nothing, and cannot be wrong, because they are your words rather than a summary of them. They also do not count against the twelve, and they still answer when the allowance is gone or when the answer service is unreachable.

Since 4.4.8 none of that needs a key. The assistant appears with or without a Groq key, answers everything your own words can answer, and when a question falls outside them says plainly that the detail is not in this shop's own words and gives the buyer your support address. With no key nothing a buyer types leaves your site at all, and the small print under the box says exactly that; with a key saved the small print says instead that the question is sent to Groq. The buyer is always told which of the two is happening, so nobody is told a question goes to Groq on a site that has no key.

Only what your facts do not cover costs a call to the outside service. If that call fails the buyer is told plainly that the detail is not in the store's own words, and your support address is offered after that sentence, not instead of an answer. The buyer is never shown your technical trouble: a refused key or a busy service reads as could not answer that one right now, and the real reason is attached to the box for you to read while you are logged in.

When a question falls outside those facts it says plainly that it does not have that detail and gives your support address. That is deliberate. A sales assistant that invents a refund window or a delivery date wins one order and costs you the argument that follows, so it is told in as many words never to guess, never to alter a price or discount, never to promise a date, and never to claim it has done something. It answers questions. It cannot take an order, and it is told never to ask for a card number or any personal details.

A bubble or a box

Since 4.4.4 it arrives as a chat bubble: a small round button at the bottom right of the checkout that opens the same form. The button is the shape people already know how to open, so it gets read; it opens upwards, clear of the pay button, and on a phone it becomes a sheet across the width instead of a card hanging off the edge. It closes on Escape, and on a tap in the corner.

The setting is labelled Checkout help bubble (4.4.8), with one sentence under it: it puts the assistant in a round corner button, instead of a box inside the cart. That is all it does.

Untick Checkout help bubble in Settings and the form sits in the cart under the total instead, the way it did from 4.4.0. There is only ever one of them: the bubble replaces the box rather than joining it, and a page with no checkout on it never gets a bubble floating over it.

Either way it is the same assistant, the same nonce, the same twelve questions an hour. The bubble is a delivery, not a second door in.

What you need

- The switch in Settings, ticked by default. Untick Ask about this order and it is gone from the page, bubble and box alike. No key is needed for anything above.
- A Groq API key in Settings is optional, the same free one the AI recovery emails use. With it, the questions your own facts do not answer are answered too. Without it the assistant simply stops at your own words, which is a working assistant, not a broken one.
- A support address in Settings is optional. Leave it empty and the WordPress admin address is used, shown in the field as a placeholder. Fill it in and that address is the one a buyer is given whenever the assistant cannot answer, on the box and in every reply that ends that way.

What it costs and who can see it

One short call per question, at most two hundred tokens. Groq is asked to answer in two sentences, so a question costs a fraction of a cent, and on the free tier nothing at all.

Twelve questions an hour per visitor address, which is more than a buyer needs and less than a scraper wants. On the thirteenth it says so and gives your email instead.

With a key saved the box tells the buyer, in small print under it, that the question is sent to Groq to be

answered; with no key it says instead that answers come from this store's own details and policies and that nothing typed there leaves the site. Both ask the buyer never to type card details. Nothing else about the buyer leaves the site: their name, their email and their card never reach the AI. Only the question they typed, plus the item facts above.

What it looks like

It is built from the same --evc-* variables as the rest of the checkout, so the design panel governs its colours, corners, spacing and type. There is nothing separate to style.

Page builders

Everything can be placed from a page builder. All three routes call the same code, so a block, a widget and a shortcode behave identically.

- Gutenberg: five blocks in the EVC group (Products, Product, Buy Button, Checkout, Field). Previews are rendered by the server, so the editor matches the page.
- Elementor: one EVC Checkout widget, with a product grid or checkout form switch and a column count. It only registers when Elementor is active.
- PageLayer, Divi and Beaver Builder: use the Shortcode, Code or HTML module. EVC -> Insert builds the shortcode from your products and copies it.

Why not a PageLayer widget: PageLayer has no public API for third-party widgets, so any plugin that pretends to add one breaks on their next update. The shortcode route renders the same markup and needs no code.

Insert (EVC → Insert)

A shortcode builder for the builders that only accept shortcodes. Pick what you want, and the box below writes the code with a Copy button.

- Product grid: columns 2, 3 or 4, and a switch for the description on each card.
- Checkout form: no options, give it its own page, usually /checkout/.
- One product, and the buy button with custom button text.
- A product field: prints one of the text custom values from Invoice Ninja, chosen by slot.

The products in the dropdowns come straight from Invoice Ninja, cached for five minutes.

The checkout flow

1. A buy button sends the visitor to the checkout page with ?product_id=123 (and ?r=CODE if an affiliate ref is in the URL).
2. The plugin creates a PaymentIntent server-side, the price is re-fetched from Invoice Ninja and never trusted from the browser.
3. The card form (Stripe Elements) confirms the payment on your domain, the customer is never redirected to stripe.com.
4. The Stripe webhook marks the invoice PAID in Invoice Ninja.

The checkout page embeds a per-user, time-sensitive WordPress nonce, and the plugin automatically excludes that page from page caching.

The buyer confirmation box

A second box on the checkout page, under the policy agreement, holding a statement you write yourself. It exists for the things a buyer has to have understood before paying that are not policies: a host setting the product needs, a minimum version, a delivery window, a size limit.

Set it in EVC, Settings, The checkout page, Buyer Confirmation Box. There is no switch to find: the words are the switch, so there is nothing to leave half set and no way to print an empty box over the pay button.

What the buyer must tick before paying
: The blank field itself. Plain text, line breaks kept, links not allowed. Write it and the box appears; empty it and the box is gone.

Also ask which hosting company they use
: One short required field under the statement, for the case where the answer decides whose support desk gets the call.

What to call that box
: The label on that field. Defaults to Hosting company.

How it behaves:

- The buyer cannot pay until the tick is in, and the check runs on the server as well as in the browser. A tick that only exists in JavaScript is evidence of nothing.
- The statement, the moment it was ticked and the hosting company (when you asked for one) are recorded three times over: on the Stripe payment, in the invoice private notes in Invoice Ninja, and in the buyer receipt email.
- Blank means no box. Nothing is shown and nothing is required.
- It is separate from the policy agreement on purpose: the policies are a lawful agreement, this is your own requirement, and the page reads them as two different things.

Write what the buyer would otherwise say they were never told, in their words rather than yours. For a plugin that needs a host extension switched on first:

```
I understand this plugin needs the ionCube Loader extension switched on
by my web host before it will run, and that this is free and my host can
enable it.
```

Webhook (required)

```
https://YOUR-DOMAIN/wp-json/evc/v1/stripe-webhook
```

In the Stripe dashboard, create a webhook endpoint at the URL above, subscribe it to the checkout.session.completed AND payment_intent.succeeded events, then paste the signing secret (whsec_...) into EVC → Settings → Stripe Webhook Secret. This is what creates the paid invoice in Invoice Ninja after the customer pays.

Renamed in 4.3.21: the REST namespace is now evc/v1 (was tim/v1). If you configured a webhook against the old tim/v1 URL, update the endpoint in the Stripe dashboard, this is the one manual step of the upgrade.

Abandoned Carts

When a customer types a valid email on the checkout page, their billing details and product are captured immediately, before any payment attempt. If payment never happens:

- One reminder email is sent about an hour later (a link straight back to their checkout with the affiliate ref intact). With AI Recovery Emails enabled (Settings, Groq API key), the reminder is written per-cart by AI in a professional sales tone; the fixed template is the automatic fallback if the API is off or fails.
- When the customer pays, their cart entries flip to Converted automatically (via the Stripe webhook).
- Manage everything under EVC → Abandoned Carts: open / abandoned / converted counts, recovered revenue, Resend reminder and Delete actions.
- A summary widget appears on the WordPress Dashboard, and a tile on EVC → Overview.

Settings → Abandoned Cart Follow-up toggles the automatic reminder email (on by default). Captured emails are stored in the evc_abandoned_carts database table.

Registry Keys (digital products)

Tick Require Reg Key on a product row in EVC → Products and that product gets a registry key fulfilled automatically when a payment succeeds. (The box used to say just Key, which read like a licence field rather than a thing the buyer receives.)

- One unused key is claimed from the stockpile, a key scoped to that product first, then the general pool.

- The key is marked used with the customer's name, email, product, invoice and timestamp.
- The key is stamped onto the Invoice Ninja invoice as custom_value1, and emailed to the customer in a dedicated message plus the purchase receipt.
- The invoice's public notes and that email also carry the version and the two addresses a buyer needs: the download, and the manual. Set them once on the product row under Download, Manual and Version, then they follow every sale. Left blank, Manual is worked out from the download address, because EVC hands a product out as /evc-download/<name>/ and its manual as /evc-download/<name>-manual/.
- The manual rides with the email as an attachment, so nobody has to log in anywhere to receive what they paid for. A copy is also filed on the invoice as a document, which is a record rather than the way the customer gets it. Nothing depends on the client portal. Anything over 5 MB is left as the link instead of being bounced by a mail server.
- The purchase email is plain text on purpose: the key, the version, the download address, the manual address, the invoice number, and the manual attached. No HTML to be stripped, broken, or filed as spam.
- None of this is tied to how EVC sells. The three fields are ordinary addresses and a version number; a product with no download address gets no guessed manual and no missing pieces in its email, and a failed fetch leaves the sale and the key alone.
- If the stockpile is empty, the site owner is emailed immediately and the sale still completes, no lost orders.

Managing the stockpile (EVC → Registry Keys)

Adding: paste one key per line and press Add keys. There is no format to obey, a key is free text up to 255 characters, so an EVC-H key, a serial from a supplier or anything you can retype all behave the same. The page answers with the honest count, 200 key(s) added, 0 duplicate(s) skipped, and repeating a paste adds nothing, because a key already in the stockpile is skipped rather than doubled. Reserved for decides the scope: a single product, or Any product.

A key reserved for a product is only ever handed to a buyer of that product. A key in the general pool can go to anyone. When a sale lands the claim takes a key reserved for that product first, then one from the general pool, oldest first.

If the wrong product is picked, nothing breaks and nothing is lost. Those keys simply never get claimed for anything else, and they show up as reserved for the wrong product rather than under the one you meant to stock. Two ways to put it right, neither of which retypes a key: Move keys moves every available key from one reservation to another in one action, and leaves keys already sent to a customer alone; the Set control on an unused key’s row re-reserves that one key on its own.

Finding things: the stockpile filters by status and by reservation and says how many keys match, so a batch that went into the wrong place can be found rather than scrolled for. The table holds the newest 300; the filter is what narrows it.

What the numbers mean: the page opens on a card for each product that asks for a key. The big number is how many a buyer of that product could actually be handed, the keys reserved for it plus the keys in the general pool, since a sale of that product can draw on either. Beside it, in smaller figures, is what is reserved for it, what is sitting in the general pool, and how many have already gone to customers. A card turns red once that product falls under your water line, and a line underneath names anything that is low.

The cards are built from the stockpile itself, so every product with keys reserved to it has one, ticked or not, in the catalogue or not. A pile nothing could ever spend is drawn in amber with a line saying so, either because the product is not ticked Require Reg Key or because it has left Invoice Ninja: its keys are still there and still reserved, and a reservation that exists is never hidden. One card at the end, All keys, carries the whole stockpile; it turns red when nothing at all is available.

Each card also says which family of key the pile actually holds, as EVC-C or EVC-H or a mixture with counts. Keys are minted per product, so the letter says what a batch was made for. If a batch was added under the wrong reservation, that line shows it in the pile it landed in rather than leaving it to be found on a row further down the page.

How the figures add up: a product’s big number counts the shared general pool, so the cards add up to more than All keys. All keys is the keys reserved to some product plus the general pool, counted once, and the line under the cards prints both halves of that sum.

Repairs: Release returns a used key to the stockpile and clears the assignment (refunds); Set re-reserves one that has not been used yet; Delete removes it for good.

Knowing before you run out

Running out is discovered at the worst moment otherwise: a buyer pays and the key has to be sent by hand. So the count is watched, per product, not as one total that can look healthy while the product people actually buy has nothing reserved for it.

- The WordPress Dashboard widget names each key product and how many a buyer of it could be handed, in red when that is under your water line.
- The Registry Keys page names the products that are under the line and how many each has left.
- You are emailed once the first time a product falls under the line, and once more when it reaches none. Both messages name the product, say how many are left and how many are in the general pool, and say where to add more. Neither repeats on later sales, because a warning that repeats stops being read.
- The water line is set on the Registry Keys page, five by default, and can be changed at any time. Add keys and the count goes back above it, which is also what lets a later fall speak again.
- The count is what a buyer of that product could really be handed: keys reserved for it plus the general pool, which is what a sale claims from. Keys reserved for a different product are never counted, because they can never be handed over for this one.

Upgrading from tim_ builds (4.3.20 and earlier)

- Saved settings migrate automatically: on the first admin load after upgrading, every old tim_* option is copied to its evc_* name.
- Old shortcode tags keep working as aliases, existing pages never render blank.
- The Stripe webhook URL changed (see above).

Troubleshooting

Fatal error on activation, another active plugin declares the same function names. EVC Checkout uses only the evc_ prefix (since 4.3.21); an old "Invoice Ninja" bridge plugin declaring tim_* functions can still collide with the legacy aliases, deactivate it (EVC replaces it).

Buy buttons do nothing / checkout says "being configured", the Stripe keys are missing or empty in settings.

"Session expired" on checkout, a caching layer is serving the checkout page. The plugin sets DONOTCACHEPAGE automatically; if your host ignores it, exclude the checkout page from cache manually.

"I pasted my key and Registry Keys is not there", that copy is the free package, which does not include the paid modules. The key is fine and is accepted there; install the paid package over this copy, using the Download the paid package link on EVC → Settings → Licence (the receipt email carries the same link), and use the same key in it. EVC → Settings → Licence names the package in the Package row.

"EVC Checkout is not running yet", and the notice names the ionCube Loader. The host has not switched the extension on. Nothing was changed and no setting was lost. Either tick ioncube_loader under cPanel, Select PHP Version, Extensions yourself, or press Copy the message for your host on the notice and send it to their support, then reload. EVC → Settings → Licence shows the package and the Loader version once it is running.

Products not showing, check the Invoice Ninja URL, API token, and company id. The product list caches for 5 minutes; the Products screen has a refresh link.

Changelog

```
4.6.7          Every Buy button now opens this product's own page with its
        product id in the address, so it lands on the thing being sold
        rather than the front of the shop.

        The link to the old paid-package download is gone from the
        plugin altogether. There is no file behind that address any
        more, so it was a dead end, and while a file did sit there it
        was a public door to the package meant to be bought.

        A copy that already carries the paid package is told where its
        key is, and is never offered the package again.
```

```
4.6.5          A package whose program files this server cannot run now
        switches itself off instead of taking the site down. On 15 Nov
        2026 evcapps.ca answered a blank 500 on every page, wp-login.php
        included, with every file in place. The guard asked one
        question, "is an ionCube Loader loaded?", the answer was yes, so
        it required the program files, and that Loader refused them. A
        refused file is a fatal at compile time and cannot be caught, so
        nothing printed and nothing said why. Turning one plugin off did
        not help, because the same mismatch was in three of them.

        The question is now "can this server run this bytecode",
        answered before the first require. The build writes
```

includes/ENCODED-FOR-PHP.txt with the PHP the files were encoded for, and the main file compares it with the running PHP using ionCube's own rule: a Loader runs its own version and the two before it. A package built for 8.1 on a 8.4 server, or one with no Loader at all, now loads, sees the mismatch and stops, leaving the site up and one notice on the Plugins screen.

The notice also stops giving the wrong advice. A Loader that is running but too new for the bytecode is not a missing Loader, and telling someone to "enable the ionCube Loader" sends them round in circles. That case now names the two versions and asks for a package encoded for their PHP, with the copy-button message for the host.

4.6.4 A paid module on a copy with no key is now switched off, not just announced. Registry Keys drew its whole screen with a notice on top, so the notice was a sentence and the screen was the lock that was not there: keys could be added, moved, released and deleted without a licence. Abandoned Carts was the same, and its reminder emails could be sent. Both screens now show the same card the free edition shows, and the work behind them refuses too, because a card on a screen is not a gate: a request can be aimed straight at the handler.

The hourly abandoned-cart reminder run stops on an unlicensed copy, so a site that lapses does not keep mailing its customers.

Found by the seller on his own test install: "if it's part of the paid system how come I can put a key in it". The honest answer was that it could.

4.6.3 A seller who has bought is no longer sold the same thing twice. On a copy that already carries the paid package, a locked module said "is a pro feature ... or get a key", and the card under it offered the paid package they were looking at. It now says the module is part of the paid edition and that the license key is in the receipt email, with the Settings screen one click away. The free edition keeps the offer, because there it is the truth.

Which copy this is comes from the package itself, not from a setting, so the wording follows the files rather than a switch that can be left half set. Both prompts read it from one place, so they cannot disagree with each other.

4.6.2 The plugin now needs PHP 8.2 rather than 8.1, and says so in its header, in its own notice and in this manual. This is not a tidy-up: the paid package's program files are ionCube bytecode, and an ionCube Loader only runs the bytecode of its own PHP version and the two before it. Their own release note puts it plainly: "the PHP 8.4 Loader will run files produced by the V15 PHP 8.4 encoder as well as files produced by the PHP 8.2 and 8.3 encoders". PHP 8.1 is not on that list. A package encoded for 8.1 therefore installs on a 8.4 site and then dies on activation with nothing on screen, which is exactly what happened when this release was tested.

The floor is 8.2 rather than 8.4 because 8.2 covers 8.2, 8.3, 8.4 and 8.5 from one package, while 8.4 covers only 8.4 and 8.5. The online Encoder carries one PHP version per job and cannot bundle passes the way the desktop Encoder does, so 8.2 is the widest single target. If you are on 8.1 and cannot move, say so through the support address and we will deal with it.

The build script refuses a download that still carries an evaluation expiry. The free Test encoding job stamps its files with the time they stop working, and that stamp is the only place the bytes say which job made them. Finishing one with the normal command used to be possible, and the result was a package that worked in testing and was dead code for a buyer. It is now refused unless you say the run is a rehearsal with --test.

The build script also refuses a download finished for more than one PHP version, because one online job cannot cover two, and it names 8.2 as the default target instead of 8.1.

4.6.1 The install step now names both packages by the name on the zip you actually downloaded: evc-checkout.zip from the free download, evc-checkout-pro.zip from the paid order. Before this it said evc-checkout.zip only, which is not the file a paid buyer holds.

The build script's file names now follow one rule across the family, so a build cannot be mistaken for something it is not: evc-checkout.zip is the free edition and the name it goes up under on the server, evc-checkout-plain.zip is the working copy with every paid module in it readable and never uploaded, and evc-checkout-pro.zip is the paid package. Until this the working

```
        copy was also called evc-checkout.zip, the same name as the file
        on the server.
```

```
4.6.0         The header now says Requires PHP 8.1 rather than 8.0, and the
        notice a too-old site sees says 8.1 too. This was a real trap
        for the paid package: its program files are ionCube bytecode
        built for PHP 8.1 and newer, so a 8.0 site would have taken the
        package, then been refused by the Loader with nothing on screen
        to explain it.

        The build script compares that header line with the lowest PHP
        the Encoder was told to cover, and stops the build when the
        header promises an older PHP than the bytecode can run. The same
        guard went into EVC Helpdesk at 1.6.1.

        The build script now refuses a package made by a trial Encoder.
        The trial's files are for testing only and stop working 36 hours
        after they are made, which would leave a buyer's site working
        today and dead tomorrow with nothing in it to say why. It reads
        what the Encoder itself says about the files and quotes it back.
        --trial-ok builds one anyway, for testing only, and the closing
        line then says it must never be installed, uploaded or sold.

        The encoded build bundles a copy for every PHP 8.x Encoder on
        the machine instead of a fixed 8.1 and 8.2 pair, so a site on
        8.3 or 8.4 is covered by the same package. It also refuses a
        file that came back from the Encoder still carrying its own
        function names, which is what a header written over copied
        source looks like.

        Every package is now checked after it is written: no test
        harness inside, the family gear mark inside, and the manual
        stamped with the same version as the plugin. The manual records
        its own version inside the file, so the check still runs on a
        Windows machine with no pdftotext, which was where it used to
        switch itself off.

        This release's manual was rebuilt, so the copy inside the
        package is the manual for 4.6.0. The last two packages carried
        the 4.5.8 manual.

        The plain build says out loud what it is: readable PHP, for
        testing only, with the encoded build named as the one that
```

ships.

4.5.9 The Metatag card on the shelf names a licence key instead of a price. No screen, manual, readme or package in this plugin prints a figure, so a price changes once in the checkout and everything follows without a release. Same pass as EVC Metatag 1.11.2, AI Builder 1.12.1, Helpdesk 1.5.2, Snippets 1.5.1 and Download Tracker 1.0.11.

4.5.8 A locked screen in the free edition no longer looks like a fault. Registry Keys and Abandoned Carts now show a faded preview of the screen behind one panel: what it does in a line, and a single I have a key button. The old bare sentence on a blank page is gone, and those screens now carry the logo and the tab row like every other.

Invoices, Products and Send Invoice on a fresh install, before Invoice Ninja is connected, now show a Connect Invoice Ninja panel with one button to the right Settings block, in place of a red error over an empty page.

The Overview's Abandoned carts tile read counts the free edition does not have, which printed PHP warnings on a fresh install. It now says Paid edition and links to what the module does.

4.5.7 A buyer confirmation box you write yourself, and nothing more complicated than that. Settings, The checkout page has one blank field: write a sentence in it and the sentence is printed on the checkout page under the policy agreement as a box the buyer cannot pay without ticking. Empty the field and the box is gone. There is no second switch, because a switch is one more thing to leave half set.

It is for whatever a buyer must have understood before the money leaves their account when it is not a policy: a host setting the product needs, a download to remember, a delivery window. The words are yours, so nothing here is our guess at what your buyers need told.

Optional second field asking which hosting company they use, with its own label. When the answer decides whose support desk gets the call, the buyer has already told you.

The statement, the time it was ticked and the host they named are recorded three times: on the Stripe payment, in the invoice's private notes in Invoice Ninja, and in the buyer's receipt email. Enforced on the server as well as in the browser, because a tick that only lives in JavaScript is evidence of nothing.

4.5.6 The link a keyed seller is given for the paid package now goes through the download tracker rather than naming the file inside /downloads. The tracker counts the download and keeps the folder out of sight, so moving the file on the server no longer breaks the link.

4.5.5 The free copy can now finish its own upgrade. It used to say "see evcapps.ca" and tell a keyed seller the paid package was "linked in your receipt email", which left the purchase path resting on a sentence in an email being correct. Paid-module screens now carry a real download link for the paid package and a real buy link, and a valid key in the free copy shows a Download the paid package button with the one sentence that matters: install it over this copy, paste the same key, nothing in your database moves.

Both addresses sit in two small functions, evc_buy_url() and evc_paid_download_url(), filterable so a reseller can point them elsewhere without editing the plugin.

4.5.4 The free download no longer contains the paid code. Switching a paid module off with a licence check was never protection, because a free download then handed over Registry Keys, Abandoned Carts and the recovery emails; those modules are now cut out of the free package altogether and the paid package is the only copy that carries them. The free package is the same source, cut by markers at build time, so there is one codebase to keep and two packages out of it.

A paid screen in the free copy is never blank and never pretends. It says the module is part of the paid edition, that this copy does not include it, and that a licence key is still read here and switches the module on in the paid package.

Your key works in both packages, and that is now proven rather than assumed: the test mints a real key, takes the key-checking

code out of each package and runs it, and the two answers must match. A mistyped key and another product's key are both refused. A licensed free copy says so on the Licence row, and points at the paid package in the receipt email.

The build refuses to produce a free package that still contains a paid region, or a full package that claims to be the free edition, or a free package that does not parse. Every PHP file in the free zip is syntax-checked before the zip is written.

4.5.3 The program files now ship encoded. Everything that does the work (the checkout engine, the shop assistant, digital delivery, the insert and render path, the blocks) goes out as ionCube bytecode rather than readable PHP, so a copy cannot be opened, copied, edited or resold. The plugin's main file stays readable on purpose, because WordPress has to read its header to list the plugin at all.

Bytecode is run by the ionCube Loader, a free PHP extension most cPanel and LiteSpeed hosts already have switched on. Where it is not switched on the plugin no longer half-loads: the Plugins screen shows one notice naming the extension and asking for it to be enabled, and says that nothing was changed and no setting was lost. There is no white screen.

Settings, Licence now carries a Package row: which package is installed, and which ionCube Loader version is running it. A support call starts with that answer instead of guessing.

The build can now produce both packages. The encoded one is built from a copy, never from the working tree, and the build reads the encoded files back: if the output does not carry the ionCube signature the build refuses rather than ship plain source under a protected label.

4.5.2 A reservation with no tick can no longer go missing from the page. The cards were built from the products ticked Require Reg Key, so 200 keys reserved to Helpdesk had no card at all while All keys still counted them: the page read 2515 keys available, 2398 reserved, and three cards adding up to 2198. Every reservation that exists now has a card, ticked or not, in the catalogue or not.

A pile nothing can spend is drawn in amber and says so in words, either that the product is not ticked to require a key or that it has left the catalogue, and one line above names every such pile, because those keys will never be handed to anyone from here.

Each card says which family of key the pile actually holds, EVC-C or EVC-H or a mixture with counts, so a batch added under the wrong reservation shows itself instead of sitting quiet.

A line under the cards explains the arithmetic: a product's figure counts the shared general pool, so the cards add up to more than All keys, and All keys is what is reserved plus the general pool counted once.

Low-stock alerts only watch products a sale can actually claim from, so a stranded pile cannot email about stock nothing spends.

A key product with no Download address on its Products row is now named on the Registry Keys page, on its card and in one line above the stockpile. Before, such a product looked ready, sold, took a key out of the stockpile, and emailed the buyer a key with nothing to download. It is only flagged while it is ticked and still has keys to hand out.

The fields on the Registry Keys page (Reserved for, Move keys, Low-stock alerts, Filter) are each set in a cell of their own, and the cells alternate light and darker across the row. The Stockpile table bands its columns the same way, so the eye can follow a key across to its status and its buyer.

Settings is grouped by subject instead of grown a row at a time: Licence, Stripe, Invoice Ninja, The checkout page, and Emails and receipts. All three Stripe entries now sit together in one block, including the webhook secret, which used to be the last row on the page under the AI panel; Invoice Ninja's three fields are together, and Company ID is no longer three rows away from the rest of it.

Each block on Settings says whether it is filled in, as Connected / Not set up for Stripe and Invoice Ninja and Licensed / Free edition for the licence, and a Jump to line at the top takes you straight to a block. A Save button sits at the top as

well as the bottom, so a long page does not have to be scrolled to be saved.

4.5.1 The Registry Keys page now opens on a row of cards, one per key product, the way the Download Tracker has always shown its counts. The big number on a card is how many a buyer of that product could actually be handed, and beside it sits what is reserved for it, what is left in the general pool, and how many have already gone to customers.

A product under your water line turns its card red, and the All keys card does the same when the stockpile has nothing available at all.

When no product asks for a key yet, the page says where to tick Require Reg Key instead of leaving an empty panel.

4.5.0 The Registry Keys page now answers where you asked it. Adding keys, releasing one or deleting one used to end on WordPress's own “The link you followed has expired” screen if the page had been open a while or the browser restored a cached copy. It now comes back here with one plain line saying what happened and that nothing was lost.

Adding keys no longer re-posts the batch when the page is refreshed, and the product you picked in Reserved for is still chosen when you land back, so a second batch for the same product does not have to be re-picked.

The stockpile can be filtered by status and by reservation, and says how many keys match, so a batch reserved for the wrong product can be found instead of scrolled for.

Move keys: when a batch was reserved for the wrong product, move every available key from one reservation to another in one action. Keys already sent to a customer are left alone.

A single key that has not been used yet can be re-reserved on its row with Set, without deleting and retyping it.

Key display: the card counts the general pool separately from what is reserved for each product, a product that has left the catalogue is named as such rather than shown as a bare id, and

its stranded keys stay visible.

Low stock is now said out loud, per product. The dashboard widget names each key product and how many a buyer of it could actually be handed, in red when it is under the line, instead of one total that could look healthy while the product people buy has nothing reserved.

The first time a key product falls under the water line, and again when it reaches none, the site admin gets an email naming the product, how many are left, and where to add more. Once per fall, not once per sale.

The water line is yours to set on the Registry Keys page (5 by default) and can be changed at any time.

4.4.8 The assistant no longer needs a Groq key to appear. Without one it answers from your item name, price, description and published policies, tells the buyer plainly when a question is outside those, and gives them your support address. Nothing a buyer types leaves the site when there is no key, and the box says so.

Support address buyers see: a new field in Settings, defaulting to the WordPress admin address. This is the address a buyer is given when the assistant cannot answer, on the box and in every reply that ends that way.

The bubble's setting is now labelled Checkout help bubble, with one sentence under it.

Test the assistant answers without a key too, using your own words, instead of complaining about a service you never signed up for.

4.3.65 The author line on the Plugins screen reads Elite Venture Capitol and nothing else.

4.3.64 The version a buyer is handed is read from the file behind the Download address instead of a number typed on the product row, so a forgotten edit cannot advertise last month's build against this month's zip.

Read once and remembered for half a day; if the file cannot be

reached the row's own number is used, so a sale never depends on the check.

The product row now says where its number came from: read from the file, or using this row because the file could not be read.

4.3.63 The manual is attached to the purchase email itself, so a buyer gets the file without logging in to any portal. The copy filed on the invoice is now the record, not the delivery.

Anything over 5 MB is left as a link rather than bounced by a mail server.

Nothing in the delivery depends on how EVC uses it: ordinary addresses, a version, and a blank field that stays blank.

4.3.62 The manual is attached to the paid invoice as a PDF document in Invoice Ninja, so the buyer has the file rather than a link to chase. A failed fetch leaves the tracked link on the invoice and never affects the sale.

The purchase email is sent as plain text on purpose: key, version, download address, manual address, invoice number. No HTML.

4.3.61 Every sale now hands over the version along with the key, on the Invoice Ninja invoice and in the email, plus the two addresses the customer actually needs: the download and the manual.

Three fields on the product row: Download, Manual, Version. Left blank, the manual is worked out from the download address as <name>-manual, because that is how EVC hands its products out.

Blank stays blank. A product with no download address is never given a guessed manual link.

4.3.60 Manual (PDF) button in the header of every EVC screen, and Manual plus More from EVC in the row under the plugin name on the Plugins screen.

4.3.59 The More from EVC tab names one product as the natural partner (EVC Download Tracker: selling a file means counting the

downloads) before listing the rest of the workshop.

4.3.58 The Download PDF button on the Overview screen now opens the manual through the EVC download tracker (evcapps.ca/evc-download/checkout-manual/), so manual downloads are counted like every other EVC download and the PDF can be replaced without a plugin release.

The copy shipped inside the plugin is still the fallback, so the button works on a site with no internet or if the tracked address is ever blanked.

docs/check_manual.py still passes (12 checks).

4.3.57 The registration key now prints on the invoice. It was saved to Invoice Ninja's custom_value1 only, and a custom field appears on the PDF only when the invoice design prints that field, so on most designs the key the customer paid for was missing from the one document they look at.

The key is written to public_notes as well, which every design prints. Existing notes on the invoice are kept and added to, never replaced, and a replayed webhook cannot write the key twice.

A rejected invoice update is now logged. The previous call discarded its result, so a failed write looked exactly like a successful one.

Send Invoice no longer sends an empty invoice number. An explicit empty string tells Ninja the number is blank rather than asking it to number the invoice, and the invoice arrived with no number on it.

New test_evc_invoice_key.php (13 assertions) covers the note builder: blank invoice, notes preserved, key appended once, replay safe, empty key changing nothing, both fields written, and no empty number sent.

4.3.56 Fixed the unreadable product card: the name, blurb, "Learn more", description body and price were hardcoded light grey (#9d9db3, #c7c7d9) and white (#ffffff), colours drawn for a dark card. On the default Light theme they sat on a white card, so

the text was nearly invisible.

Every card colour now comes from the Design theme: var(--evc-text), var(--evc-muted), var(--evc-line) and var(--evc-accent). Nothing on the card is painted for one theme, so Light, Dark and every preset stay readable.

The card border uses the theme accent instead of a written-out #0b85f8, and the divider above "Learn more" and the product image frame use the theme line colour instead of translucent white.

The buy modal's fallback colours were the dark palette, so a page without the design CSS got white text on a white panel. They now fall back to the light palette, matching the default theme.

Design preset accents and the remaining #0b85f8 literals move to the brand blue #0f87f8.

New test_evc_card_colors.php (28 assertions) renders real cards and fails on any literal text colour other than the green Buy button, checks every variable the card uses is declared by the design tokens, and measures WCAG contrast for both palettes (all four pairs pass AA 4.5:1).

4.3.55 Product cards carry a short description on the face: the first paragraph of the Ninja Notes field, cut on a word boundary at 150 characters and clamped to three lines, so a row of cards stays level.

The full notes stay behind "Learn more" as before. One description in Invoice Ninja, used twice at two lengths.

[evc_products excerpt="90"] shortens the blurb, excerpt="0" removes it. Persona landing pages use the same blurb under the tagline.

A product with no notes gets no blurb rather than an empty gap, and show_description="open" skips the blurb because the first paragraph is already on the card.

The cut counts characters through pcre rather than mbstring, so

an accent or an em dash is never sliced in half on a PHP build without mbstring.

Fixed the version stamp drift: EVC_VERSION said 4.3.53 while the plugin header said 4.3.54.

4.3.54 Product categories, built from Invoice Ninja: Settings › Product Categories points at one of the product's custom values, and [evc_products group="category"] renders one headed section per category while category="Name" shows just one. Doubles as the copy for [evc_products] pages.

Fixed before it could happen: persona detection now skips the category slot. Both helpers scan custom values, so a product with "Plugins" as its category would have been served to the site as persona "Plugins".

Grouping is alphabetical with uncategorised products last under "Other", so a partly filled catalogue never leads with a blank heading.

Nothing changes for an install that leaves categories off, the shortcode behaves exactly as before.

4.3.53 Design is now the only place appearance lives. Checkout Theme, Accent Color, Card Background Override and Custom CSS are gone from Settings, which carries a single button that opens the Design tab.

Custom CSS moved onto the Design screen and saves with the same button as the tokens. Still Unlimited-license: typed and kept on a free install, printed only with a key.

The four appearance options are no longer registered with the Settings form, so a plain Settings save can no longer blank a colour the design panel never posted.

4.3.52 Finish the CSS refactor: the checkout markup reads the design tokens itself instead of carrying its own numbers, so corners, padding, spacing and type stay in step with the panel.

Tokens are declared on :root and inherit, so they reach the buy pop-up and the success panel too. 63 hardcoded values replaced,

with the old value kept as the fallback in every case.

4.3.51 One row of tabs across every EVC screen: Overview, Products, Invoices, Send Invoice, Abandoned Carts, Registry Keys, Design, Insert, Settings.

Each tab is a plain link that reloads the screen, so the Settings form and the Products table keep their own submissions and nothing on the checkout path was re-plumbed.

4.3.50 Page builder support: five Gutenberg blocks (product, price, buy button, checkout form, field) and an Elementor widget, all calling the same renderer as the shortcodes.

New EVC -> Insert screen: pick a product, get the shortcode or block for your builder, with a copy button. Exists because PageLayer, Divi and Beaver have no public API for third-party widgets.

Fixed on the way: the Field block lost its alignment attribute, and scripts resolved their path one directory short.

4.3.49 Number not used. Nothing was released as 4.3.49; the tab row shipped as 4.3.51.

4.3.48 New Design page: five presets, six colours, corner radius, padding, spacing, image height, type face, heading weight, shadow, button style and button width, with a live preview and a per-preset reset.

Design tokens print as one small style block (about 700 bytes) of CSS variables plus scoped rules, with no build step and no extra request.

Sites that never open the panel keep the exact palette they had; the legacy Theme, Accent and Card Background options still drive it.

EVC_VERSION now matches the plugin header (it had drifted to 4.3.43).

4.3.47 Buyer consent on every purchase: an agreement checkbox before

paying, recorded on the Stripe payment with which policy set was agreed and the exact time.

Built-in policy set (Terms, Privacy, Acceptable Use) opens in a pop-up, so there are no pages to create and nothing to write. A site can point at its own pages instead.

4.3.46 Shipping address can differ from billing: off for digital goods, optional via a "Ship to a different address" checkbox, or always asked.

The address is saved to the Stripe payment and to the Invoice Ninja client's shipping fields. Billing stays billing.

4.3.45 Opening the checkout page without a product no longer dead-ends on a page nothing created: it now shows the store's own product list under a "Choose what you'd like to buy" heading.

An invalid or unpurchasable product gets a short notice and the same list. The /pricing/ page with [evc_products] is created automatically if missing, because every historical link points there.

4.3.44 Fix: a fresh install landed on a blank checkout page. The Buy button points at the configured URL, but the form is the [evc_checkout] shortcode and nothing ever created that page.

Two guards: a missing page at the configured path is created (activation, per-version admin pass, or when the path setting changes, parents included), and an existing page that lacks the shortcode gets the checkout rendered onto it anyway. The page stays excluded from caching.

4.3.43 Unified EVC family key structure (EVC-C-XXXX-XXXX-XXXX); older keys without the product letter are still honoured.

4.3.42 License gate extended: one Unlimited key now unlocks unlimited products plus the pro modules (Abandoned Carts, AI Recovery Emails, Registry Keys, Custom CSS).

Unlicensed installs keep the full core checkout with a 5-product limit; gated admin pages and widgets show an upgrade notice

instead of erroring.

4.3.41 New: AI Recovery Emails. Abandoned-cart reminders written per-cart by AI (Groq), professional sales tone, no invented discounts or urgency.

Falls back to the fixed template when the toggle is off, the key is missing, or any API call fails, so reminders are never lost. Settings: AI Recovery Emails toggle plus Groq API key.

4.3.40 New: Registry Keys. Products ticked Key in EVC -> Products pull one key from the stockpile on payment: marked used (who, when, invoice), stamped on the Invoice Ninja invoice as custom_value1, and emailed to the buyer.

New EVC -> Registry Keys admin page (bulk add, scoped or general pool, release/delete, low-stock warnings), Dashboard widget, and an owner email when the stockpile runs dry.

4.3.39 New: Abandoned Carts. Billing details are captured at checkout, an hourly job sends one follow-up email about an hour later with a link straight back, and a successful payment marks the cart Converted.

New EVC -> Abandoned Carts admin page (stats, resend, delete), WP Dashboard widget, Overview tile, and a toggle in Settings (default on).

4.3.38 Fix: card descriptions never rendered. Cards now show About <product> behind a Learn more / Show less toggle, and show_description="open" renders it expanded.

New: Custom CSS in Settings, loaded after the checkout theme, overriding every --evc-* variable.

4.3.37 Admin menu icon: custom EVC logo replaced with the built-in WordPress shopping-cart icon (dashicons-cart).

4.3.36 Admin menu renamed EVC to EVC-Checkout, menu title and logo alt text included.

4.3.35 Fix: [evc_field] threw a fatal error from a leftover TIM_CACHE_SECONDS constant. The product cache now uses EVC_CACHE_SECONDS (5 minutes).

4.3.34 keygen --log appends issued keys to keys-issued.txt with a date stamp; Customer column fix on Invoices.

4.3.33 Fix: the Customer column on admin Invoices always showed a dash. The client relation is now requested, with a cached per-customer fallback lookup.

4.3.32 Fix: paid invoices showed as Cancelled. The status labels were off by one against Invoice Ninja's codes; the correct mapping is 1 Draft, 2 Sent, 3 Partial, 4 Paid, 5 Cancelled, 6 Reversed.

4.3.31 Fix: the PDF button on admin Invoices showed raw PDF text instead of downloading the file. The download now runs before WordPress prints the admin page.

4.3.30 New high-resolution EVC logo with a light sidebar icon variant, used on the admin header, the manual cover and page headers.

4.3.29 The legacy tim_ compatibility layer is gone from the distribution build: the six [tim_*] shortcode aliases, the options migration and the Overview footnote.

4.3.28 EVC branding restored: wordmark header on all admin pages, custom sidebar menu icon, stale EVC_VERSION constant fixed.

4.3.27 Edition licensing: free is 5 products, an EVC key unlocks unlimited. License field in Settings, Locked badges on Products, edition status on the Overview page, gates on every buy path.

4.3.26 Themeable checkout: Light/Dark theme, accent colour, card background override with transparent supported.

4.3.25 Overview admin page: build, product count, shortcodes, manual. [evc_checkout] added to the Products table.

4.3.24 White-label pass: RAM checkbox, support-year upsell, persona-

```
        module metadata, ResidenceAI wording removed.

4.3.23          Removed the license-server ownership guard: settings check, live
        email check, Core auto-include and double-buy block.

4.3.22          Restored [evc_field], documented in the manual and on the
        settings page.

4.3.21          Full tim_ to evc_ rename, with an automatic settings migration
        and legacy shortcode aliases so it cannot collide with an old
        tim_ plugin.

4.3.20          White-label cleanup: ResidenceAI trial signup and persona
        landing shortcodes removed.

4.3.19          First cleaned white-label build replacing the old prototype.

4.6.6          The locked screens now show the real thing, greyed: Registry
        Keys and Abandoned Carts draw their own columns and four sample
        rows, faded and unpressable, so you can see what the paid
        modules do before you buy them. They used to show anonymous grey
        bars, which told you something was there but not what.

        The repository builds two folders from one source: a free
        edition and a paid edition. Zip the one that is going on the
        site.

        Nothing is encoded any more. The program files are plain PHP in
        both copies, so there is no ionCube Loader to ask your host for,
        no expiry, and no package that can take a site down because a
        server cannot read it.

        The manual no longer describes the encoded package, because
        there isn't one.
```