

ETECHFLOW • MAGENTO 2 / ADOBE COMMERCE • HYVÄ + LUMA

# Extra Options Plugin

Reusable product-option templates • radio-card + conditional inputs • multi-file upload • bulk price • theme-adaptive

A custom-options power-up for Magento 2 / Adobe Commerce. Build option SETS once as reusable “Option Templates” — radio / checkbox / dropdown choices with conditional cut-code and image-upload sub-fields — then apply them to any product or whole category in one click. The storefront renders them as clean, theme-coloured radio cards that work on both Hyvä and Luma, with soft-required file handling and a live price that updates as the customer chooses.

**Etechflow\_OptionsPlugin • Module Documentation & User Guide**

Magento / Adobe Commerce 2.4.6 – 2.4.9 • PHP 8.1 – 8.4 • Hyvä & Luma • Part of the eTechFlow Suite

© eTechFlow • [license-service.etechflow.com](https://license-service.etechflow.com)

# Table of Contents

1. Overview
2. Core Concepts — Option Templates
3. Architecture & How It Works
4. Installation & Setup
5. Configuration Reference
6. Building an Option Template
7. Applying Templates to Products & Categories
8. The Storefront Experience
9. Admin Tools — Bulk Price & Migration
10. Licensing & Activation
11. Troubleshooting / FAQ

## 1 • Overview

### 1.1 What the module does

Magento's built-in **Customizable Options** are powerful but painful: you rebuild the same choices on every product by hand, there's no way to reuse a set across a catalog, and file-upload options block the cart when left empty. **Extra Options Plugin** fixes all of that. You build an option **set once** — a reusable **Option Template** — with radio / checkbox / dropdown choices and, where you need them, conditional sub-fields (a text box for a cut code, an image upload for a photo). You then **apply that template to any product, or to an entire category**, and the module syncs the underlying Magento custom options for you.

On the storefront the same options render as clean, **theme-coloured radio cards** that work on **Hyvä and Luma** alike, with a live running price and soft-required file handling so an empty upload never blocks checkout.

#### Works on Hyvä AND Luma

Unlike many option extensions, this module is **theme-adaptive**. It reads the active theme's accent, text and border colours at runtime and styles the option cards to match — so it looks native on Luma, Adobe Commerce, Hyvä or a custom theme, with no hard-coded palette. The screenshots in this guide are from a live Hyvä store (Keystation).

### 1.2 Highlights

- **Reusable Option Templates** — build a set of choices once, apply it to many products or a whole category.
- **Radio-card UI** — a primary radio/dropdown option (e.g. “Would you like us to cut this key?”) with related inputs is grouped into a single, tidy card.
- **Conditional sub-fields** — show a **cut-code text box** or an **image upload** only when the relevant choice is selected; validate it only then.
- **Multi-file image upload** — up to **10 files** per option, for customers who need to send several photos.
- **Soft-required files** — a file option left empty never triggers Magento's hard “required option(s) weren't entered” error, so the cart isn't blocked.
- **Apply to a category** — link a template to a category and every product in it inherits the options.
- **Bulk Price Update** — change option prices across many products from one admin page.
- **Migration Tool** — import existing keyword-based option config into templates, so you don't rebuild by hand.
- **Theme-adaptive appearance** — auto-adopts the live theme's colours (Hyvä / Luma / custom), with an optional accent override.
- **Licence-gated admin** — the Templates / Bulk-Price / Migration tools require a valid eTechFlow licence; the storefront keeps working.

### 1.3 Compatibility matrix

Area	Supported
Magento / Adobe Commerce	2.4.6 – 2.4.9 (Open Source & Commerce)
PHP	8.1 – 8.4
Storefront theme	Hyvä and Luma (theme-adaptive)
Built on	Magento native Customizable Options (catalog_product_option)
Licensing	eTechFlow license-service (SP key / HMAC / Bundle key)
Package	etechflow/module-options-plugin (Etechflow_OptionsPlugin) v1.0.3

## 2 • Core Concepts — Option Templates

The one idea to understand is the **Option Template**. Everything else follows from it.

Term	What it means
<b>Option Template</b>	A named, reusable set of product options (radio / checkbox / dropdown, with optional sub-fields and prices). Managed centrally under eTechFlow → Extra Options → Option Templates.
<b>Option</b>	One choice group inside a template — e.g. a radio “Would you like us to cut this key?” or a checkbox “Extras”.
<b>Value / Choice</b>	One selectable item inside an option — e.g. “Yes”, “Gift wrap +£3.00”. Values can carry a price.
<b>Sub-field</b>	A conditional input shown only when its parent value is selected — e.g. a “cut code” text box or an image upload under “Yes”.
<b>Link (apply)</b>	The act of attaching a template to a product or category. On save the module syncs the corresponding Magento custom options onto the product.

### Why templates beat native options

Native custom options are **per-product** — 20 products that need the same “Cut this key?” question means 20 hand-built copies, and a price change means editing all 20. An **Option Template** is built once and **linked**; change the template (or run Bulk Price) and every linked product updates. Link it to a **category** and new products in that category inherit it automatically.

## 3 • Architecture & How It Works

### 3.1 The module

A self-contained module — **Etechflow\_OptionsPlugin** (package `etechflow/module-options-plugin`). It ships an admin **Option-Templates** UI and data model, a product-form modifier, a category link, a save observer that syncs options, storefront ViewModels + templates for the radio-card rendering, an admin configuration section, and the eTechFlow licence layer. Its config lives under the **eTechFlow** tab, labelled **Extra Options Plugin** (`etechflow_options`); its admin pages live under **eTechFlow → Extra Options**.

### 3.2 How it works, end to end

1. **Build** a template in the admin Option-Templates UI — add options, values, prices and conditional sub-fields.
2. **Link** the template to a product (via the product-edit **Linked Templates** field) or to a category.
3. **On save**, a product-save observer reads the linked template IDs and a **SyncService** writes the matching rows into Magento's native `catalog_product_option` tables — so the options are 100% standard Magento under the hood.
4. **On the storefront**, ViewModels feed the template's structure (which option is the primary radio, which values have sub-fields, the checkbox mode, the theme colours) to a small set of theme-agnostic scripts that render the radio-card, wire the conditional fields, handle multi-file upload and keep the live price in sync.
5. **Licence gate** — the admin tools sit behind an eTechFlow licence check; without a valid licence the Templates / Bulk-Price / Migration pages are locked, while already-synced storefront options keep working.

### 3.3 Built on native custom options

Because the module syncs into Magento's own `catalog_product_option` tables, the options behave like any other custom option in cart, checkout, order, invoice and PDF — no bespoke cart logic to break on upgrade. The template is the **authoring layer**; native custom options are the **runtime**.

## 4 • Installation & Setup

### 4.1 Install & enable

Distributed via the private eTechFlow Composer repository; your licence key doubles as the download token. From your Magento project root:

```
composer config repositories.etechflow composer
https://license-service.etechflow.com/composer

composer config http-basic.license-service.etechflow.com you@example.com SP-XXXX-XXXX-
XXXX-XXXX

composer require etechflow/module-options-plugin

bin/magento module:enable Etechflow_OptionsPlugin

bin/magento setup:upgrade

bin/magento setup:di:compile      # production mode

bin/magento cache:flush
```

(The module can also be dropped into `app/code/Etechflow/OptionsPlugin` and enabled the same way — useful for a manual install.)

### 4.2 First-run steps

6. Paste your licence key under **Stores → Configuration → eTechFlow → Extra Options Plugin → License** (see Section 5.6 / Section 10).
7. Turn on the **master switch**: **General → \*Enable Module\* = Yes**.
8. Build (or migrate) your first **Option Template** under **eTechFlow → Extra Options → Option Templates** (Section 6).
9. Open a product and pick the template in **Linked Templates**, then **Save** (Section 7).
10. Visit the product on the storefront to see the rendered radio-card (Section 8).

#### Dev / staging domains run unlocked

Hosts beginning with `dev.`, `staging.`, `stage.`, `qa.`, `uat.`, `test.`, `preview.` or `sandbox.` are treated as development and run **fully unlocked with no licence**, so you can build and test freely. Set **License → Enforce Licence on This Domain = Yes** to validate a production key on a staging URL before go-live.

## 5 • Configuration Reference

**Stores → Configuration → eTechFlow → Extra Options Plugin** (`etechflow_options`). The section opens with a **Module Status** banner (licence + module state), then the groups below. The banner turns **green** when the module is active and the licence is valid for the current domain:

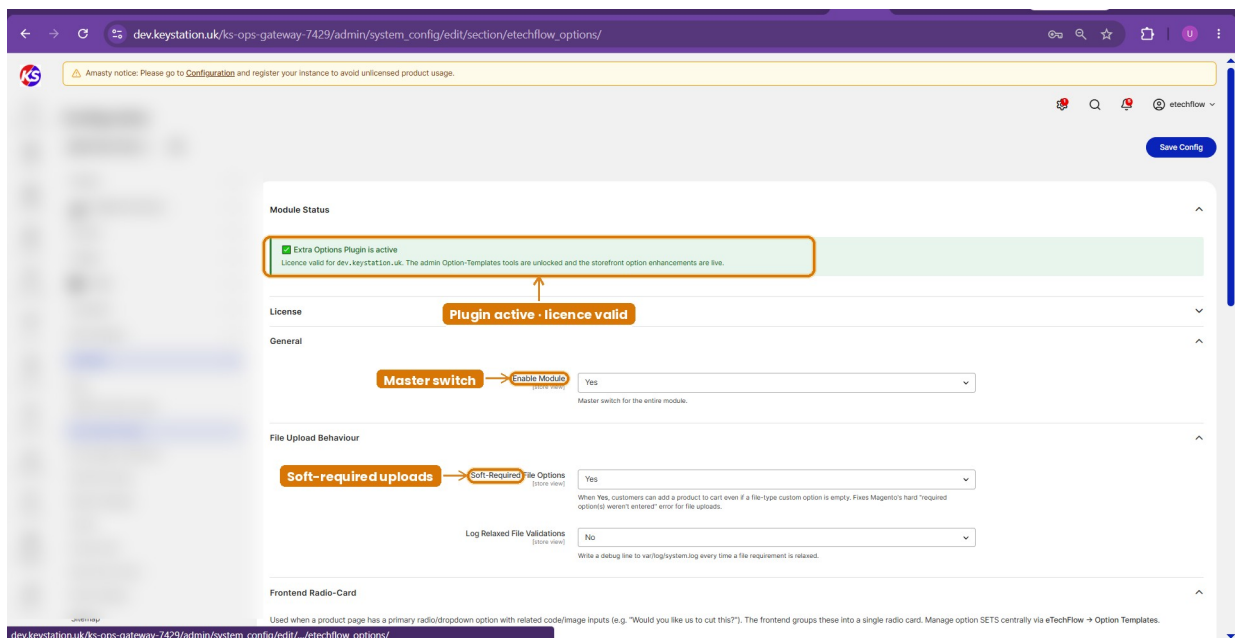


Figure 1 — the top of the configuration section — a green “Extra Options Plugin is active · Licence valid for dev.keystation.uk” Module Status banner, the License group, General → Enable Module = Yes (the master switch), and the File Upload Behaviour group (Soft-Required File Options = Yes, Log Relaxed File Validations = No).

## 5.1 Module Status

A read-only banner at the top of the section. **Green** = the module is active and licensed for this domain (the admin tools are unlocked and the storefront enhancements are live). If the licence is missing or invalid it turns red and tells you what to fix.

## 5.2 General

Setting	Default	Purpose
Enable Module (master switch)	—	Master on/off for the whole module. While OFF the storefront enhancements and admin syncing are inactive.

## 5.3 File Upload Behaviour

Setting	Default	Purpose
Soft-Required File Options	Yes	When Yes, a customer can add to cart even if a file-type option is empty — fixes Magento's hard “required option(s) weren't entered” error for uploads.
Log Relaxed File Validations	No	Writes a debug line to var/log/system.log each time a file requirement is relaxed. Only shown when Soft-Required is Yes.

## 5.4 Frontend Radio-Card

Controls the flagship storefront widget — the single card that groups a primary radio/dropdown option with its related code/image inputs (e.g. “Would you like us to cut this?”). Below it sits the Appearance group:

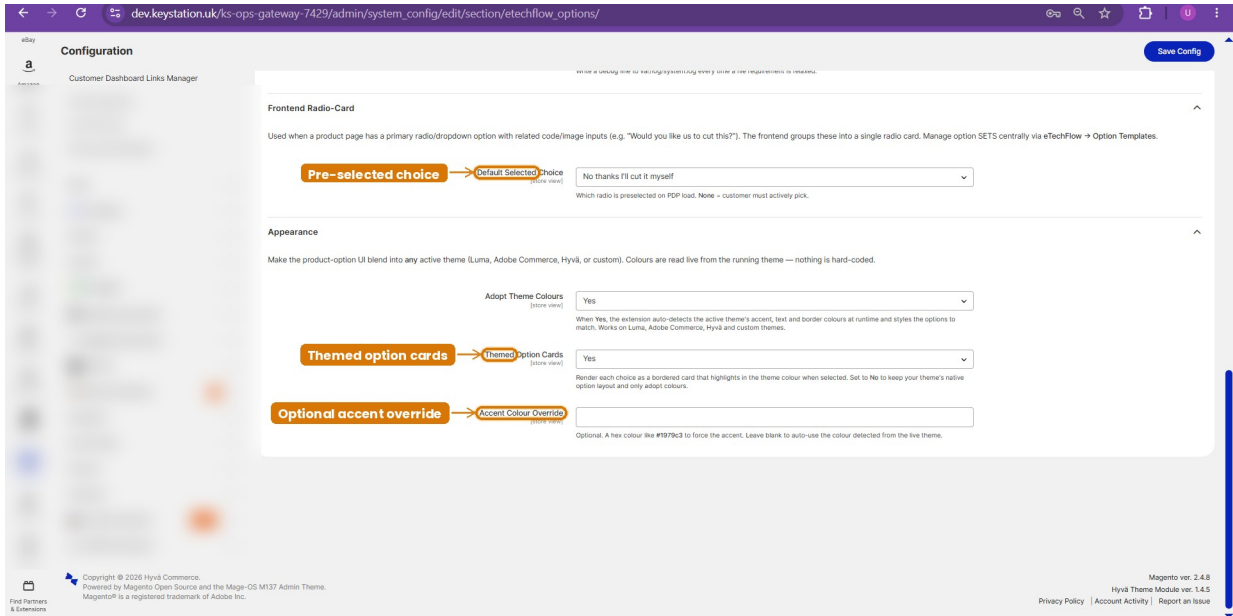


Figure 2 — the Frontend Radio-Card group (Default Selected Choice) and the Appearance group (Adopt Theme Colours, Themed Option Cards, Accent Colour Override).

Setting	Default	Purpose
<b>Default Selected Choice</b>	—	Which radio is preselected when the product page loads. None = the customer must actively pick a choice.

## 5.5 Appearance

Makes the option UI blend into **any** active theme. Colours are read live from the running theme — nothing is hard-coded — which is what lets the same module look native on both Hyvä and Luma.

Setting	Default	Purpose
<b>Adopt Theme Colours</b>	Yes	Auto-detect the active theme's accent, text and border colours at runtime and style the options to match (Luma, Adobe Commerce, Hyvä, custom).
<b>Themed Option Cards</b>	Yes	Render each choice as a bordered card that highlights in the theme colour when selected. No = keep the theme's native option layout, adopt colours only.
<b>Accent Colour Override</b>	(blank)	Optional hex colour (e.g. #1979c3) to force the accent. Blank = auto-use the colour detected from the live theme.

## 5.6 License

Setting	Purpose
<b>License Key</b>	Your SP-XXXX subscription key, or a per-module / bundle HMAC key. Bound to your domain (www. normalised).
<b>Enforce Licence on This Domain</b>	Force a dev/staging host to require a valid licence — use it to test a production key before go-live.

<b>Issued Key (auto)</b>	Read-only. Set on first successful validation; used to restore the key after an IP-block event. Do not edit.
<b>License Portal URL</b>	The eTechFlow endpoint that validates keys — default license-service.etechflow.com/license/validate.
<b>Bundle License Key</b>	One shared key that activates every eTechFlow module on the host (stored encrypted). Set EITHER this OR the per-module key.

## 6 • Building an Option Template

Everything to do with options is managed centrally under **eTechFlow → Extra Options → Option Templates**. The grid lists every template, how many options each holds, and how many products and categories it's linked to:

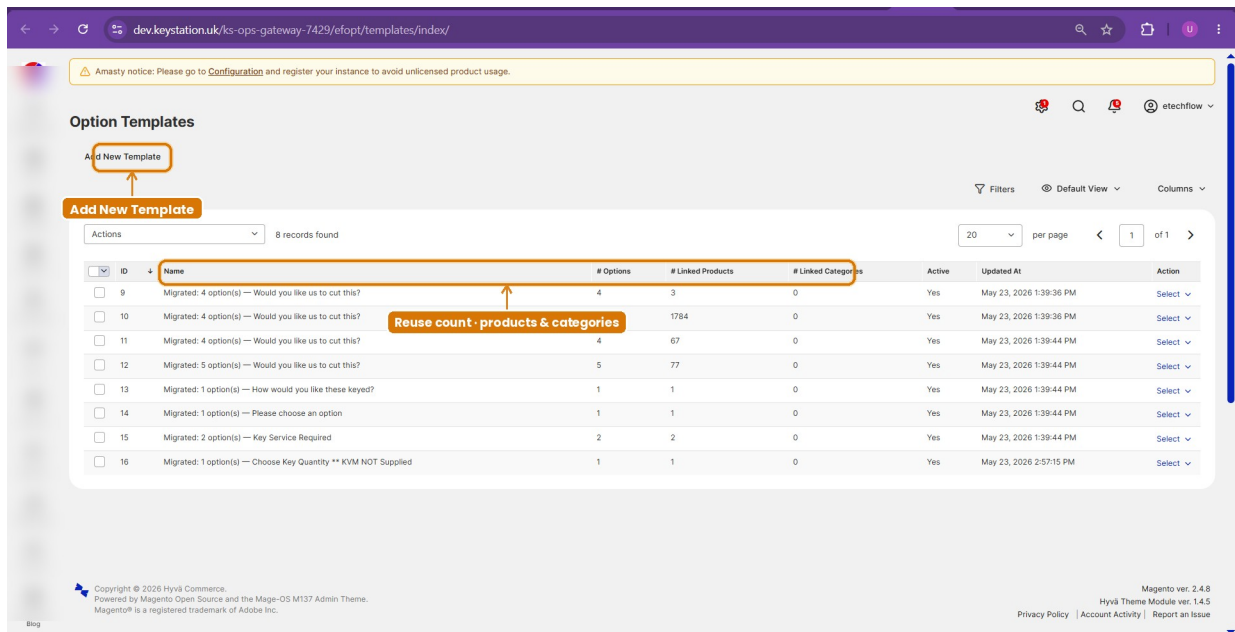


Figure 3 — the Option Templates grid — each row is a reusable template with its ID, Name, number of Options, Linked Products, Linked Categories and an Active flag, plus an “Add New Template” button. (The rows named “Migrated: N option(s)…” were imported by the Migration Tool from the store’s old keyword-based option config — see Section 9.)

Click **Add New Template** (or a row) to open the builder. Give the template a **Name** and optional **Description**, then add one or more **options**. Each option is one of **Radio**, **Checkbox** or **Dropdown**, can be marked **Required**, and holds one or more **values** (each with an optional **price**). Under a value you can attach **conditional sub-fields** — a **Text** input (e.g. a cut code) or an **Image Upload** — that only appear, and only validate, when that value is selected:

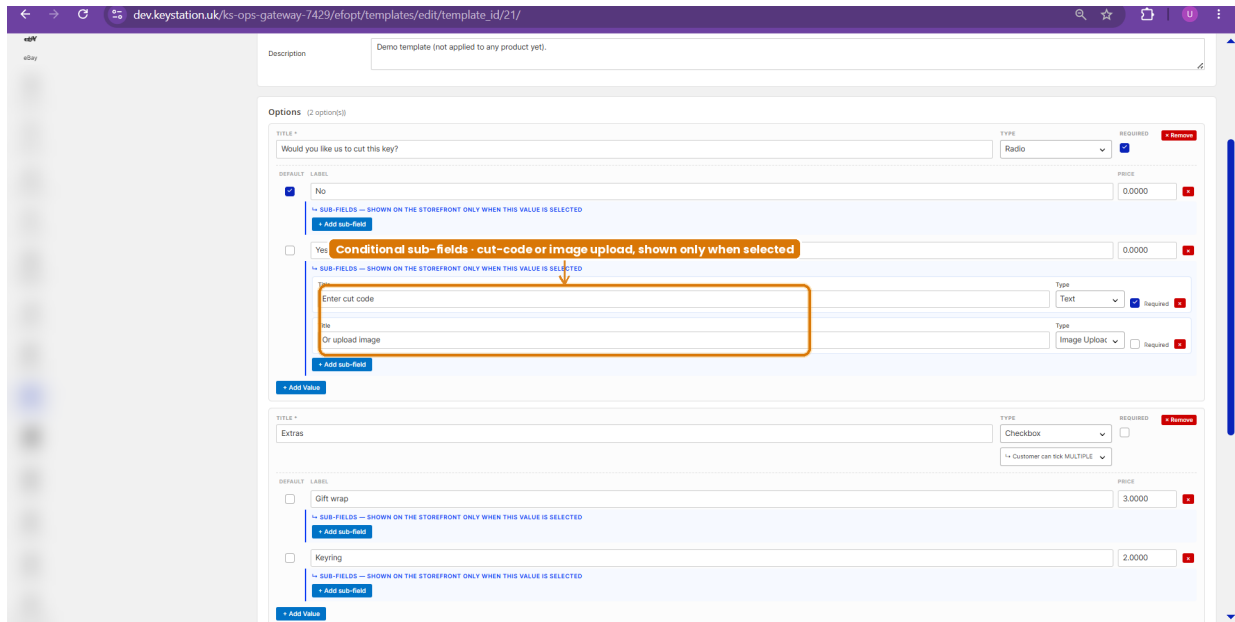


Figure 4 — the template builder for “Test Options” — Option 1 “Would you like us to cut this key?” is a required Radio with values No (default) and Yes; under Yes sit two conditional sub-fields, an “Enter cut code” Text input and an “Or upload image” Image Upload. Option 2 “Extras” is a Checkbox (“customer can tick MULTIPLE”) with values Gift wrap £3.00 and Keyring £2.00.

### Radio vs Checkbox

A **Radio** (or Dropdown) lets the customer pick **one** choice — ideal for the primary “Cut this key?” question, and it’s the option the Frontend Radio-Card is built around. A **Checkbox** lets the customer tick **multiple** add-ons independently (Gift wrap AND Keyring). Conditional sub-fields (cut code / image) hang off a specific **value**, so they surface only for the choice that needs them.

Save the template. If you also link it to products or categories from the builder’s **Apply To** picker, the module syncs those options immediately; otherwise it stays a library entry you can link from the product page.

## 7 • Applying Templates to Products & Categories

A saved template does nothing until it’s **linked**. The module adds two fieldsets to the product-edit page. The **eTechFlow Extra Options** fieldset carries the per-product toggle, and the collapsible **eTechFlow Option Templates** fieldset is where you link templates:

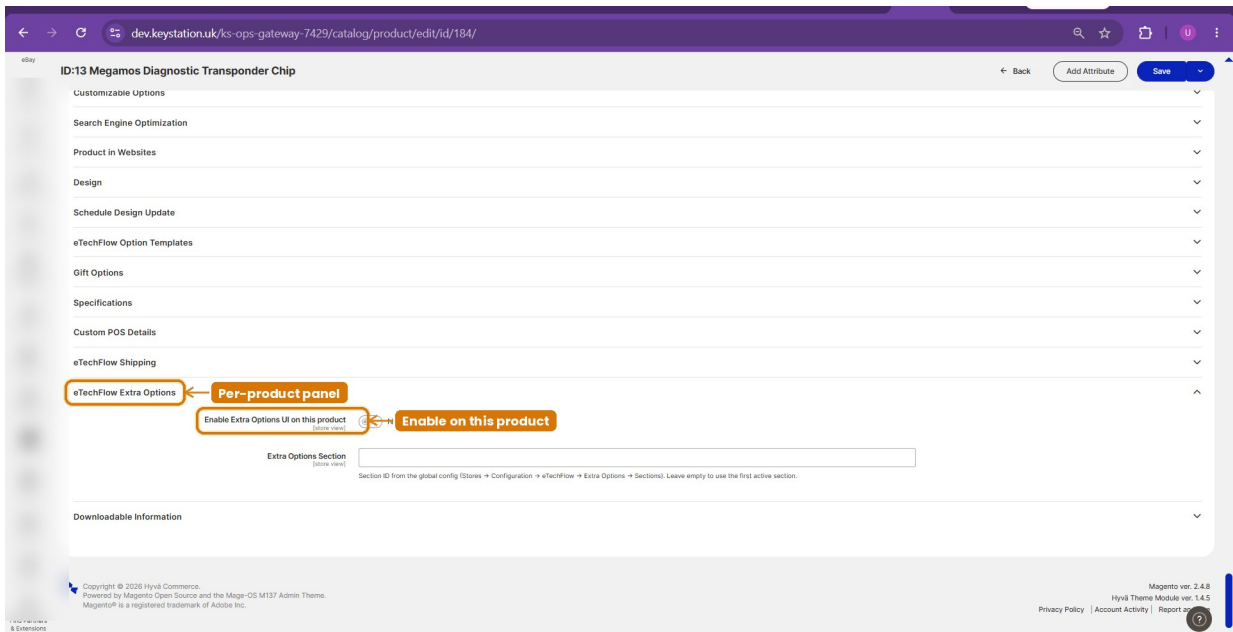


Figure 5 — the product-edit page — the collapsible “eTechFlow Option Templates” fieldset and the expanded “eTechFlow Extra Options” fieldset (the Enable Extra Options UI toggle and the Extra Options Section field).

Open **eTechFlow Option Templates** → **Linked Templates** and select one or more templates for this product. On **Save**, the options from the selected templates are pushed onto the product (re-syncing if already linked):

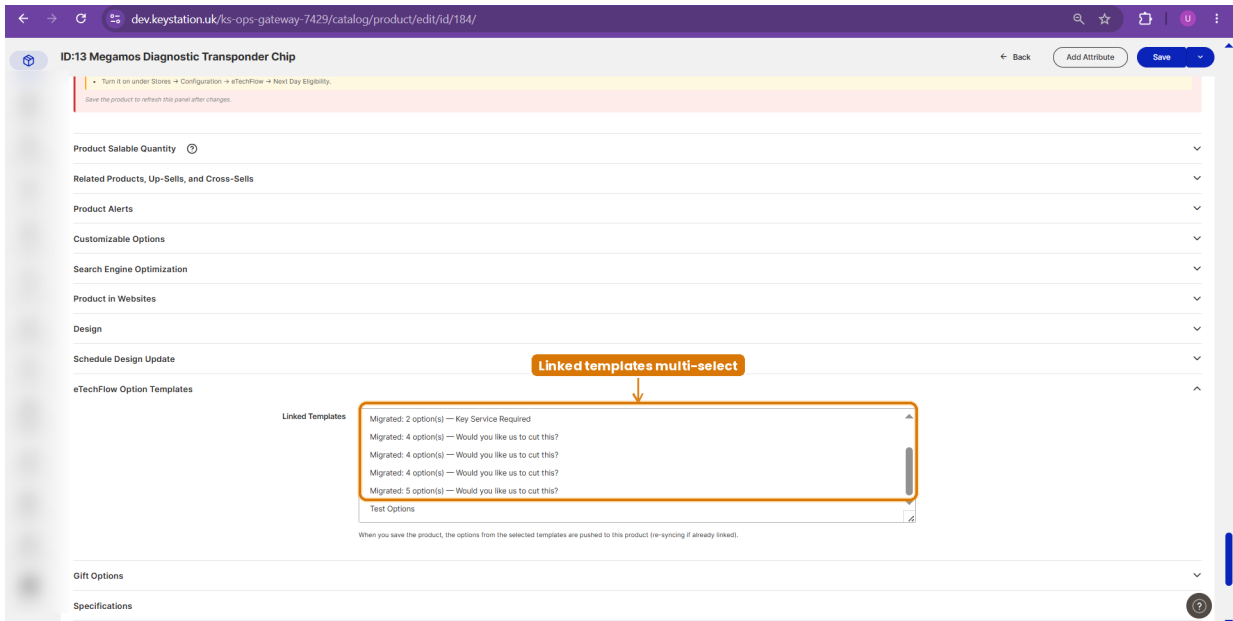


Figure 6 — the “Linked Templates” multi-select on product ID 13 (Megamos Diagnostic Transponder Chip) — the migrated templates and “Test Options” are listed, with the notice “When you save the product, the options from the selected templates are pushed to this product (re-syncing if already linked).”

### Category-wide apply

You can also link a template to a **whole category** (from the template's Apply-To picker). Every product in that category then inherits the template's options — and new products added to the category pick them up too — without touching each product individually. Category-inherited options are shown read-only on the product page so you know where they

come from.

### If a linked template doesn't appear on the product

The Linked Templates value is stored under the product form's data . product scope and read back by the save observer. If you build the module from source, that fieldset's dataScope **must** be the absolute data . product — otherwise the field renders empty and the link never saves. This is fixed in v1.0.2+.

## 8 • The Storefront Experience

On the product page the linked options render as a clean, theme-coloured card. The primary radio (“Would you like us to cut this key?”) is grouped with its conditional inputs, and the checkbox add-ons sit below — all styled in the live theme's colours, with a running price and an Add-to-Basket that isn't blocked by an empty upload:

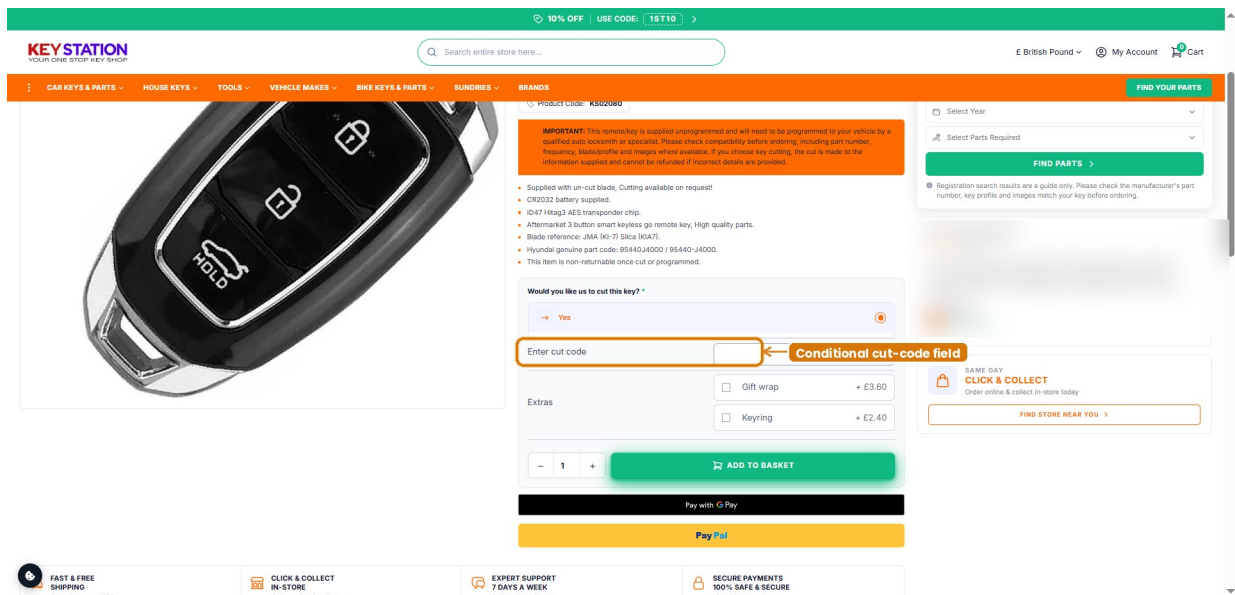


Figure 7 — the storefront product page (Hyvä, Key Station theme) with the template rendered — the “Would you like us to cut this key?” radio-card (No / Yes, Yes selected) revealing the “Enter cut code” field, then the “Extras” checkboxes (Gift wrap +£3.00, Keyring +£2.00), quantity, Add to Basket and PayPal. Every choice label renders correctly — no “undefined”.

### 8.1 What the customer sees

- **Radio-card** — the primary “Cut this key?” choice is one tidy card; picking **Yes** reveals its conditional cut-code / image inputs, picking **No** hides them.
- **Conditional validation** — the cut code / image is only required when the choice that needs it is selected, so “No” customers are never asked for a code.
- **Multi-file upload** — where enabled, the image option accepts up to 10 files.
- **Checkbox add-ons** — Gift wrap and Keyring can be ticked independently, each adding its price.
- **Live price** — the displayed price updates as choices are made.
- **Soft-required files** — leaving an image empty won't block Add to Basket (per the File Upload Behaviour setting).

## 8.2 Correct labels on every theme

Some Hyvä themes bind their own price-injection script to custom-option labels, and on certain themes that script can blank a checkbox label to the literal text “undefined” even though the server data is correct. The module ships a small, theme-agnostic **label safety-net**: it captures each correct server-rendered label and restores it if a theme ever blanks it. It's a no-op on themes that render correctly and never runs on Luma — so labels like “Gift wrap” and “Keyring” always display, as shown above.

## 9 • Admin Tools — Bulk Price & Migration

Beyond the template builder, **eTechFlow → Extra Options** provides two time-savers:

### 9.1 Bulk Price Update

**eTechFlow → Extra Options → Bulk Price Update**. Change option/value prices across many products at once from a single page — instead of editing each product's options by hand. Ideal for a catalog-wide price change on a common add-on (e.g. raising “Gift wrap” from £3 to £4 everywhere).

### 9.2 Migration Tool

**eTechFlow → Extra Options → Migration Tool**. Imports an existing **keyword-based option configuration** into proper Option Templates, so stores already using a hand-rolled or legacy option scheme don't have to rebuild from scratch. The “Migrated: N option(s) — ...” rows in the templates grid (Section 6) were produced by this tool.

### 9.3 Where they live

Admin page	Menu path	What it's for
Option Templates	eTechFlow → Extra Options → Option Templates	Build and manage reusable option sets.
Bulk Price Update	eTechFlow → Extra Options → Bulk Price Update	Change option prices across many products at once.
Migration Tool	eTechFlow → Extra Options → Migration Tool	Import legacy keyword-based options into templates.
License	eTechFlow → Extra Options → License	Activate / view the module licence.

## 10 • Licensing & Activation

Part of the **eTechFlow Suite**, activated with a key from the eTechFlow portal and validated against **license-service.etechnology.com**, bound to your licensed domain. The **admin tools** (Option Templates, Bulk Price, Migration) require a valid licence; storefront options that are already synced keep working regardless.

11. Obtain a key — an **SP-XXXX-XXXX-XXXX-XXXX** module key, or a shared **Bundle** key that activates every eTechFlow module on the host. The key is also your Composer download token.
12. Enter it under **Extra Options Plugin → License → License Key** (or set the suite-wide **Bundle License Key**), then **Save Config** and flush the cache.
13. The module validates the key against your domain (HMAC), so a key issued for one store won't activate another. The **Module Status** banner at the top of the section shows the current licence state.

### Dev/staging vs production

Development hosts (dev., staging., qa., test., sandbox....) run **unlocked with no key** so you can build freely. Turn on **Enforce Licence on This Domain** to require the key on a staging URL and confirm your production licence before go-live. A single **Bundle** key is the simplest option if you run several eTechFlow modules.

## 11 • Troubleshooting / FAQ

### **A checkbox label shows “undefined” on the storefront.**

That's a theme script blanking the label, not bad data — the module's built-in label safety-net restores it automatically (Section 8.2). Make sure you're on v1.0.3+ and flush the cache; the correct label (e.g. “Gift wrap”) will display.

### **I linked a template but the options don't appear on the product.**

Save the product after selecting it in Linked Templates — the sync runs on save. If you built from source, confirm the Option Templates fieldset uses `dataScope = data.product` (fixed in v1.0.2+); otherwise the field posts outside the product payload and the link never saves.

### **The admin Option Templates / Bulk Price / Migration page is locked.**

Those tools require a valid licence. Paste your key under Extra Options Plugin → License, Save Config and flush the cache. On a dev/staging host they're unlocked automatically unless you turned on Enforce Licence on This Domain.

### **Adding to cart fails with “required option(s) weren't entered” for a file upload.**

Set File Upload Behaviour → Soft-Required File Options = Yes. That lets customers add to cart even when a file option is empty, which is the usual intent for optional photo uploads.

### **Does it work on Luma as well as Hyvä?**

Yes. The module is theme-adaptive — it reads the live theme's colours at runtime and renders on both Luma and Hyvä (and custom themes). The label safety-net simply never triggers on Luma.

### **The option cards don't match my theme colours.**

Check Appearance → Adopt Theme Colours = Yes. To force a specific accent, set Appearance → Accent Colour Override to a hex value (e.g. #1979c3). Set Themed Option Cards = No if you'd rather keep your theme's native layout and only adopt colours.

### **How do I change one add-on's price everywhere at once?**

Use eTechFlow → Extra Options → Bulk Price Update — it changes option/value prices across many products from one page instead of editing each product.

### **I already have options set up a different way — must I rebuild?**

No. Run the Migration Tool (eTechFlow → Extra Options → Migration Tool) to import your existing keyword-based option config into templates, then refine them in the builder.

### **Which radio is selected by default on the product page?**

Set Frontend Radio-Card → Default Selected Choice. Choose None to force the customer to pick actively.