

Redirect Manager

[Luma](#) • [Hyvä](#) • [301 / 302](#) • [404 Catcher](#) • [Adobe Commerce](#)

A 301/302 redirect manager with a built-in 404 catcher — managed redirects that run BEFORE Magento's URL rewrites, with per-store rules, an active toggle, hit analytics and one-click recovery of broken URLs. FPC/Varnish-safe, managed entirely from the admin.

Etechflow_RedirectManager • **Module Documentation & User Guide**

Compatible with Magento / Adobe Commerce 2.4.6 – 2.4.9 • PHP 8.1 – 8.4 • Luma • Hyvä • Adobe Commerce

© Etechflow • license-service.etechflow.com

Table of Contents

1. Overview
2. Architecture
3. Installation & Setup
4. Admin: Managing Redirects
5. Feature 1 — Redirect Engine & Precedence
6. Feature 2 — The 404 Catcher
7. Configuration Reference
8. Licensing & Activation
9. Troubleshooting / FAQ

1 • Overview

1.1 What the module does

Etechflow Redirect Manager is a 301/302 redirect manager with a built-in 404 catcher for Magento 2 / Adobe Commerce. You manage redirects from the admin — a grid plus an add/edit form — per store view, each with an active toggle, an internal note and a live hit counter.

Its custom router runs BEFORE Magento's native `url_rewrite`, so a managed redirect takes precedence over catalog/CMS URL rewrites. The 404 catcher logs every not-found URL (with hit count and referrer) so you can turn a frequently-hit broken URL into a managed redirect in one click. Part of the Etechflow SEO Suite — a leaner alternative to the redirect tools in larger suites.

1.2 Highlights

- **301 / 302 redirects** managed from the admin — per store view, active toggle, internal notes, and a live hit counter per redirect.
- **Runs before `url_rewrite`** — a custom router (sortOrder 15) so managed redirects take precedence over catalog/CMS rewrites.
- **404 catcher** — every not-found URL is logged with hit count + referrer; one click on a 404 row creates a draft redirect to set the target.
- **Flexible targets** — the target can be a relative path ([new-page](#)) or a full external URL (<https://...>).
- **Master switch** — the whole module (engine + 404 logger) is off by default; inert until enabled.
- **Noise control** — a configurable exclude regex keeps static assets out of the 404 log.
- **FPC/Varnish-safe & licensed** — real HTTP 301/302 responses; activation via the eTechFlow licence portal.

1.3 Compatibility matrix

Area	Supported
Magento / Adobe Commerce	2.4.6, 2.4.7, 2.4.8, 2.4.9
PHP	8.1, 8.2, 8.3, 8.4
Storefronts	Luma, Hyvä, Adobe Commerce (the engine is theme-independent — it issues HTTP redirects)
Caching	Built-in Full Page Cache and Varnish — fully supported

2 • Architecture

2.1 The module

A single, self-contained Magento module — `Etechflow\RedirectManager` (Composer package `etechflow/module-redirect-manager`, namespace `Etechflow\RedirectManager`). It owns its own data, the storefront redirect router, the 404 logger, the admin grids/forms and configuration.

2.2 Runs before URL rewrites — and stays cache-safe

- **Custom router at sortOrder 15** — registered ahead of Magento's `url_rewrite` router, so a managed redirect wins even when the same path also resolves to a catalog/CMS page.
- **Real HTTP redirects** — the engine issues an actual 301/302 response (Location header), not cached page content, so it behaves correctly behind Full Page Cache / Varnish.
- **404 logging hooks the no-route page** (`controller_action_predispatch_cms_noroute_index`); 404 responses aren't FPC-cached, so every genuine miss is seen.
- **Best-effort hit counter** — incrementing a redirect's hit count never blocks or delays the redirect.

2.3 Data model

Two tables:

Table	Holds
<code>etechflow_redirect</code>	Managed redirects: <code>request_path</code> → <code>target_path</code> , <code>redirect_type</code> (301/302), <code>store_id</code> (0 = all), <code>is_active</code> , <code>hits</code> , <code>internal note</code> , <code>timestamps</code> .
<code>etechflow_redirect_404_log</code>	Captured 404s: <code>request_path</code> (unique per store), <code>hit count</code> (upserted on each miss), <code>referrer</code> , <code>store_id</code> .

3 • Installation & Setup

3.1 Install & enable

Distributed through the private eTechFlow Composer repository; your licence key is also the download token.

From the Magento project root:

```
composer config repositories.etcheflow composer
https://license-service.etcheflow.com/composer
composer config http-basic.license-service.etcheflow.com you@example.com SP-XXXX-XXXX-XXXX-XXXX
composer require etechflow/module-redirect-manager
bin/magento module:enable Etechflow_RedirectManager
bin/magento setup:upgrade
bin/magento setup:di:compile          # production mode only
bin/magento cache:flush
```

This creates the two tables and adds the admin menu under Marketing → Redirect Manager.

Important

The module ships with its master switch OFF. Nothing redirects and no 404s are logged until you set Stores → Configuration → ETECHFLOW → Redirect Manager → General → Enable Redirect Manager = Yes (or `bin/magento config:set etechflow_redirectmanager/general/enabled 1`).

3.2 Configuration

Settings live at Stores → Configuration → ETECHFLOW → Redirect Manager.

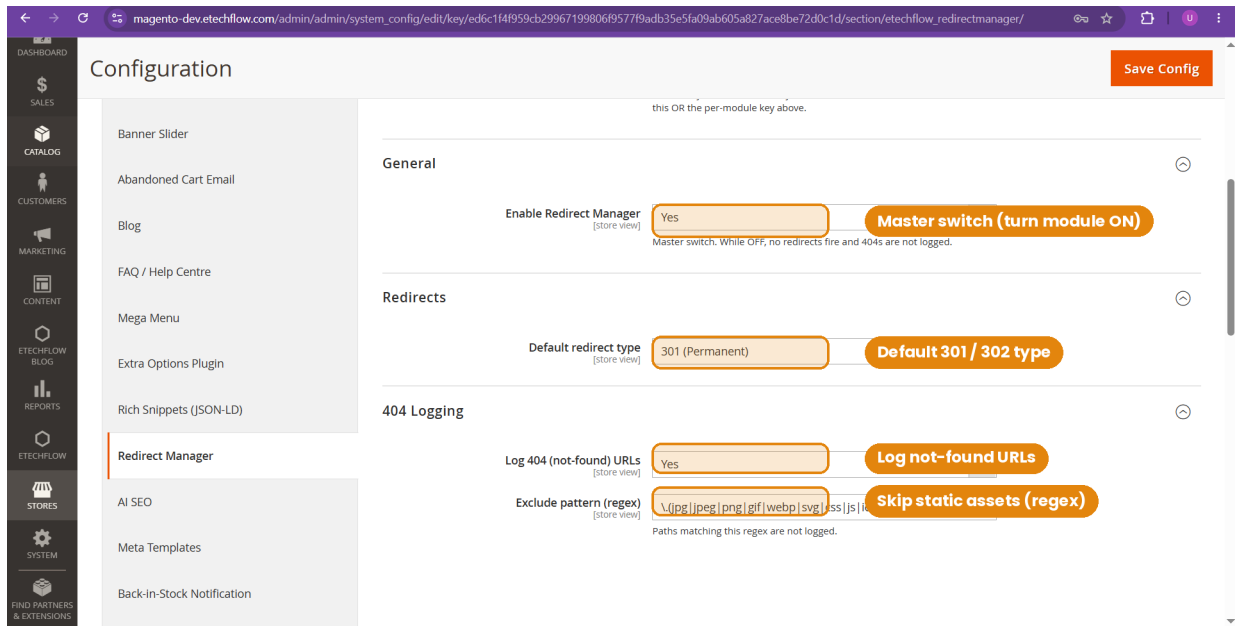


Figure 1 — Module configuration: master switch, default redirect type, and 404 logging with its exclude regex.

4 • Admin: Managing Redirects

Create and manage redirects under Marketing → Redirect Manager → Redirects. The grid lists every redirect with its paths, type, store, active state and hit count, with filtering, search and mass-delete. Click Add New Redirect (or a row) to open the form.

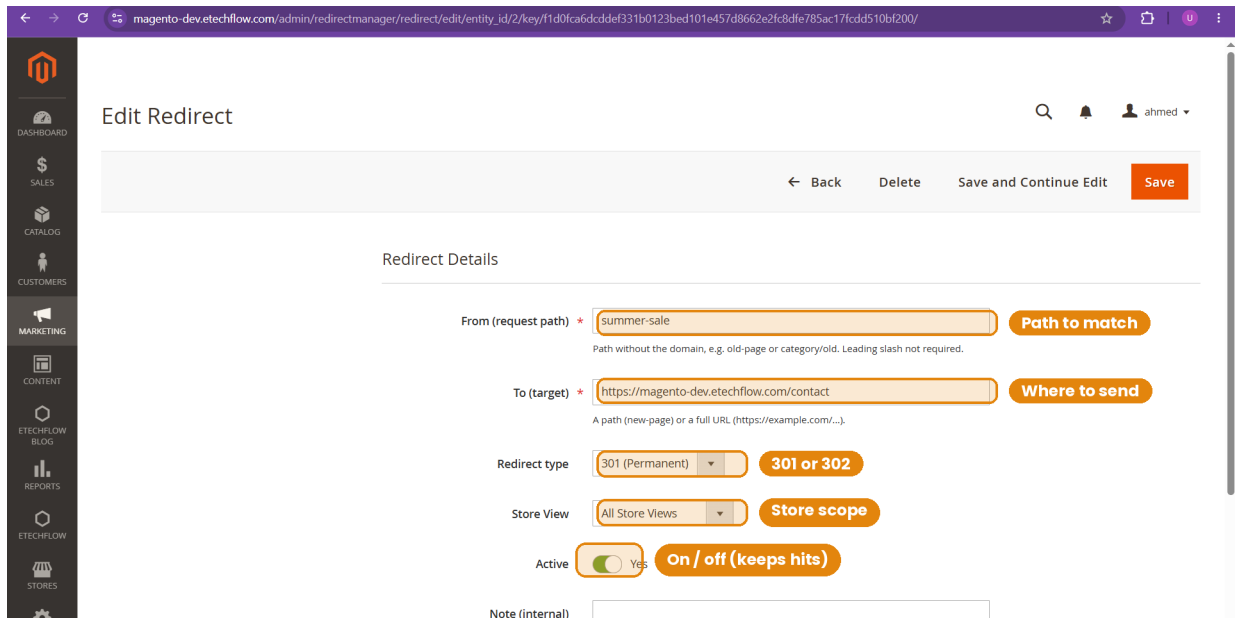


Figure 2 — The Add/Edit Redirect form.

Form fields:

Field	Meaning
From (request path)	The incoming path to match — relative, without the domain (e.g. <code>old-page</code> or <code>category/old</code>). Leading slash not required.

To (target)	Where to send it — a relative path (new-page) or a full external URL (https://example.com/).
Redirect type	301 (Permanent — passes SEO value, cached hard by browsers) or 302 (Temporary).
Store View	All Store Views (store_id 0) or a specific view.
Active	Toggle the redirect on/off without deleting it (keeps its hit history).
Note (internal)	A private note for your team; never shown on the storefront.

Important

Request Path must be RELATIVE (e.g. [summer-sale](#)) — not a full URL. The router matches it against the incoming request's relative path, so a stored <https://.../summer-sale> would never match. Only the Target may be a full URL.

FEATURE 1 OF 2

5 • Redirect Engine & Precedence

The engine matches the incoming path against active redirects for the current store and issues the configured 301/302. Because its router is registered at sortOrder 15 — ahead of Magento's url_rewrite — a managed redirect takes precedence even over a path that also resolves to a real catalog or CMS page.

- **Precedence** — a redirect with Request Path [contact](#) will fire even though [/contact](#) is a real CMS page; the managed redirect wins.
- **Path vs full-URL targets** — a relative target is resolved against the store base URL; a <https://...> target is used verbatim (off-site redirects).
- **Per store view** — a redirect set for All Store Views (store_id 0) applies everywhere; a store-specific one overrides it for that view.
- **Active toggle** — only `is_active = 1` redirects fire; deactivating pauses one without losing its hit history.
- **Hit counter** — each match increments the redirect's hit count (best-effort), so you can see which redirects are actually used.

The screenshot shows the 'Edit Redirect' form in the Magento Admin interface. The form is titled 'Redirect Details' and includes the following fields and options:

- From (request path) ***: A text input field containing 'go-google'. A label 'Relative request path' is shown to the right. Below the field, it says 'Path without the domain, e.g. old-page or category/old. Leading slash not required.'
- To (target) ***: A text input field containing 'https://google.com'. A label 'Full external https:// target' is shown to the right. Below the field, it says 'A path (new-page) or a full URL (https://example.com/...)'.
- Redirect type**: A dropdown menu set to '301 (Permanent)'.
- Store View**: A dropdown menu set to 'All Store Views'.
- Active**: A toggle switch set to 'Yes'.
- Note (internal)**: An empty text input field.

At the top right of the form, there are buttons for 'Back', 'Delete', 'Save and Continue Edit', and 'Save'.

Figure 3 — A redirect to a full external URL: the Target is an absolute https:// address (here go-google → https://google.com).

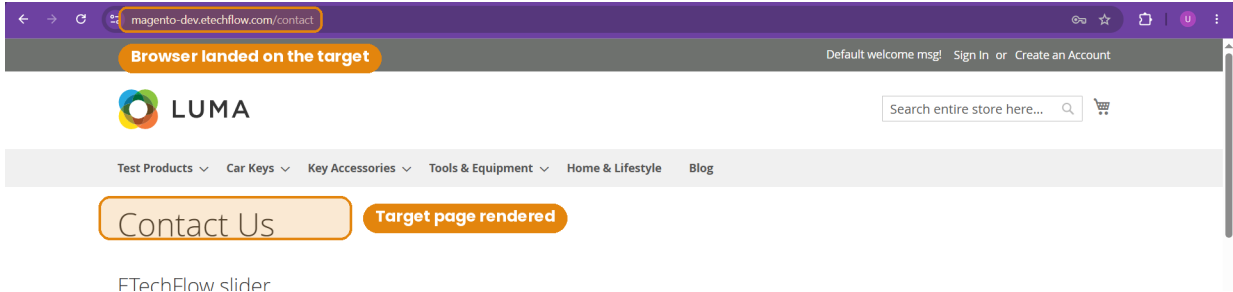


Figure 4 — A managed redirect firing on the storefront (request path → the target page).

Tip

Browsers cache 301s aggressively and the storefront is FPC-cached — always test redirects in an incognito window (and add a throwaway query like `?x=1` to force a fresh hit when checking the counter).

FEATURE 2 OF 2

6 • The 404 Catcher → Redirect Workflow

With logging on, every not-found (404) URL is recorded under Marketing → Redirect Manager → 404 Log — with a hit count (upserted on each miss, unique per store) and the referrer that linked to it. This turns your 404s into a prioritised worklist of broken URLs.

- **One-click Create Redirect** — on any 404 row, click Create Redirect: it pre-fills a draft redirect with that path and opens the form for you to set the target.
- **Exclude regex** — paths matching the configured regex (static assets by default: `\.(jpg|css|js|...)$`) are never logged, keeping the list signal-rich.
- **Backstop for URL-key changes** — Magento still writes its own 301s when you change a product/category URL key; this catcher records anything that slips through so you can convert it in one click.

Note

Both the engine and the 404 logger respect the master switch — when the module is disabled (or unlicensed), nothing is redirected and nothing is logged.

7 • Configuration Reference

Setting (path)	Default	Purpose
general/enabled	No (0)	Master switch — the redirect engine AND the 404 logger.
redirects/default_type	301	Default HTTP status for new redirects (301 permanent / 302 temporary).
log404/enabled	Yes (1)	Whether not-found URLs are logged.
log404/exclude_patterns	asset regex	A regex of request paths to skip logging (static assets / noise).
license/license_key	—	Your SP-XXXX subscription key (or HMAC key) — see §8.
license/portal_url	license-service...	The licence validation endpoint (leave as shipped).
license/bundle_license_key	—	Shared key that activates every eTechFlow module on this host.

8 • Licensing & Activation

Activated with a licence key from the eTechFlow portal, validated live against `license-service.etechflow.com` and bound to your store domain and server IP.

1. Obtain a key (SP - XXXX - XXXX - XXXX - XXXX) from your eTechFlow account.
2. Go to Stores → Configuration → ETECHFLOW → Redirect Manager → License.
3. Paste the key and save — validation runs immediately and unlocks the module.

If the module is not yet licensed, opening Marketing → Redirect Manager shows the in-admin activation gate — purchasable plan cards (Weekly / Monthly / Yearly), an Enter License Key shortcut, and a Select Plan & Pay checkout that issues a key within seconds.

Important

Full-module enforcement (fail-closed): when the licence is missing, suspended, expired, IP/domain-blocked or for the wrong domain, the ENTIRE module is disabled — no redirects fire, no 404s are logged, and the admin Redirects / 404 Log pages redirect to the licence gate. Only the gate and Stores → Configuration stay reachable so you can re-license. Validation is lightly cached (~30–60 s).

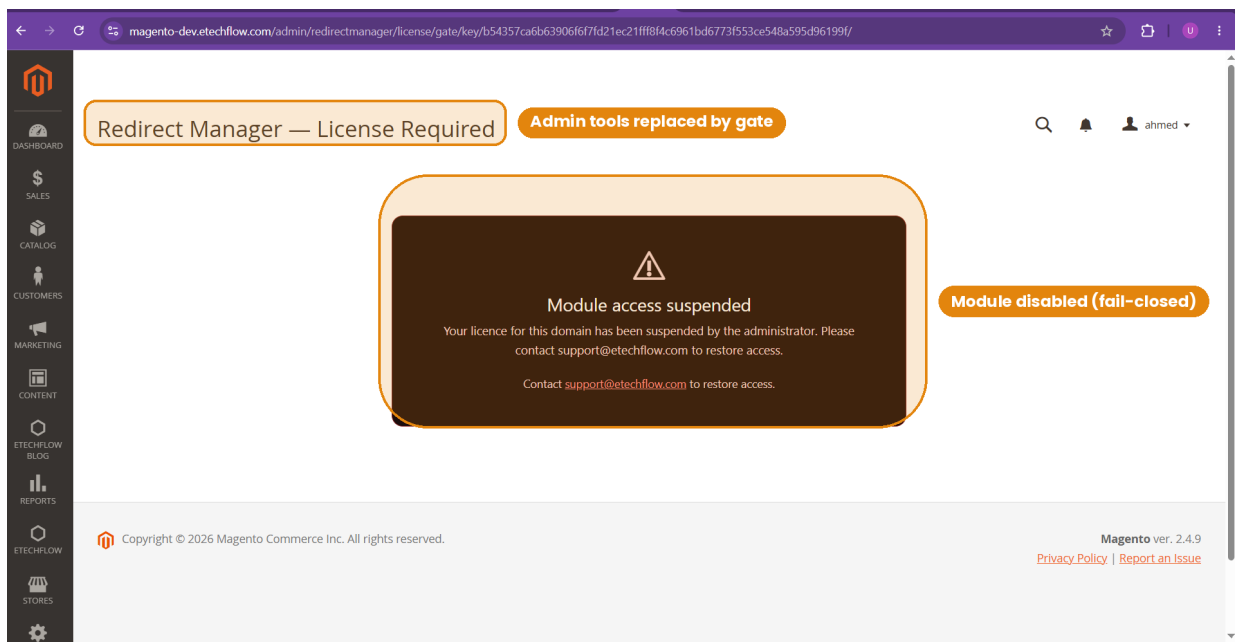


Figure 5 — A suspended licence: the admin shows a "Module access suspended" notice instead of the redirect tools.

Note

A shared eTechFlow Bundle License Key can activate every eTechFlow module on the same host — set it instead of the per-module key.

9 • Troubleshooting / FAQ

My redirect doesn't fire.

Check: the master switch is On; the licence is valid; the redirect is Active; and — most common — the Request Path is RELATIVE (`summer-sale`), not a full URL. Then flush cache and test in incognito (browsers cache 301s; the storefront is FPC-cached).

The Add/Edit Redirect form is stuck on a loading spinner.

Fixed in v1.1.1 — the form ui_component now declares templates/form/collapsible (Magento 2.4.9 doesn't render the default form template). Update to ≥1.1.1 and flush cache.

A redirect for an existing page (e.g. /contact) isn't taking over.

It should — the router runs at sortOrder 15, before url_rewrite. Flush cache; verify the redirect is Active and matches the exact relative path; test in incognito.

The hit counter isn't increasing.

A browser-cached 301 won't re-hit the server. Test in incognito with a unique query each time (`?x=1`, `?x=2`) so each visit reaches the server.

My 404s aren't being logged.

Check 404 Logging = Yes, the master switch is On, the licence is valid, and the path isn't matched by the exclude regex.

The admin Redirect Manager pages redirect to a "License" screen.

The licence is missing/suspended/invalid. Re-activate it (Stores → Config → Redirect Manager → License, or the webstore portal); the gate clears within ~60 s.

Does it replace Magento's URL Rewrites?

No — it runs ahead of them and complements them. Magento still writes its own 301s on URL-key changes; this module backstops the rest and lets you manage redirects centrally with hit analytics.