User Guide

AdminWizard for Magento 2.x — Extmag_Improver

2026-05-20

18.3 *Force Logout* button does not appear on Admin Sessions

18.4 Indexer status stays *outdated* even after pause is lifted

18.5 Categories Grid says *Class "…" does not exist*

18.6 Plugin Viewer shows duplicate `before` plugins on one method

18.7 Cron Monitor *Run Now* does nothing

18.8 Extensions List shows no *Latest version*

18.9 2FA bypass is on, but admin still sees the OTP prompt

18.10 DB tables disappeared after disabling a module

1 About the module

AdminWizard logo

AdminWizard is a toolbox of admin-side utilities that fill in the gaps in Magento 2's native back office. Every feature is reachable from a single menu under `Extmag → AdminWizard`. The bundle contains:

- **Dashboard** — single landing page with module status and quick links.
- **Hotlinks** — shortcuts to frequently used admin pages, manual or auto-generated per admin user.
- **Advanced Filters** for the Product, Order, Invoice, Shipment and Creditmemo grids — rule-based filtering on any attribute, plus customer-, address-, and item-level conditions.
- **Extensions List** — every installed module with current and latest versions.
- **Environment Info** — PHP, Magento, disk, database, Redis, Varnish, Elasticsearch status on one page.
- **Plugin & Preference Viewer** — every plugin and preference in the DI graph, with conflict detection.
- **Cron Monitor** — every cron job, last run, duration, status, and *Run Now*.
- **Config Diff** — every value overridden at website or store scope vs the default, with CSV export.
- **Admin Sessions** — every active admin session with *Force Logout*.
- **Indexer Manager** — pause, throttle, reindex, and history view per indexer, plus dependency tree.
- **Categories Grid** — flat grid replacement for the tree view, with filter / sort / mass actions.
- **Multistore Configuration** — gear icon in the system configuration to see the value of a field on every store at a glance.
- **2FA bypass** — admin-side toggle to disable Magento_TwoFactorAuth globally, in developer mode only, or for API token generation only.
- **Save DB tables on module removal** — prevents accidental data loss when a module is disabled.

- **Logs Manager Pro** and **Backup Manager Pro** — bundled at no extra cost through the `extmag/logs-core` and `extmag/backup-core` dependencies.

2 Admin navigation

Top-level menu: `Extmag → AdminWizard`.

Item	URL
Dashboard	`improver/dashboard/index`
Extensions List	`improver/extension/index`
Environment Info	`improver/environment/index`
Plugin & Preference Viewer	`improver/diViewer/index`
Cron Monitor	`improver/cron/index`
Config Diff	`improver/configDiff/index`
Admin Sessions	`improver/adminSession/index`
Configuration	`Stores → Configuration → Extmag → AdminWizard`

Cross-references in other Magento areas:

- `Catalog → Inventory → Categories Grid` — the flat grid replacement.
- Indexer Manager hooks into the native `System → Tools → Index Management` grid (adds `backlog` and `last_error` columns and a per-row detail view).
- Hotlinks render on the standard admin dashboard.

Each menu item has its own ACL resource under `Extmag_Improver::improver`, so role assignment is granular.

3 Dashboard

`Extmag → AdminWizard → Dashboard` is the recommended landing page after login. It summarises:

- Module enable status and configuration warnings.
- Auto Hotlinks shortcut row (when enabled) — the operator's own top N visited pages, click to jump.
- Quick links to the other AdminWizard features.

The dashboard is read-only — every operation lives on its own menu page.

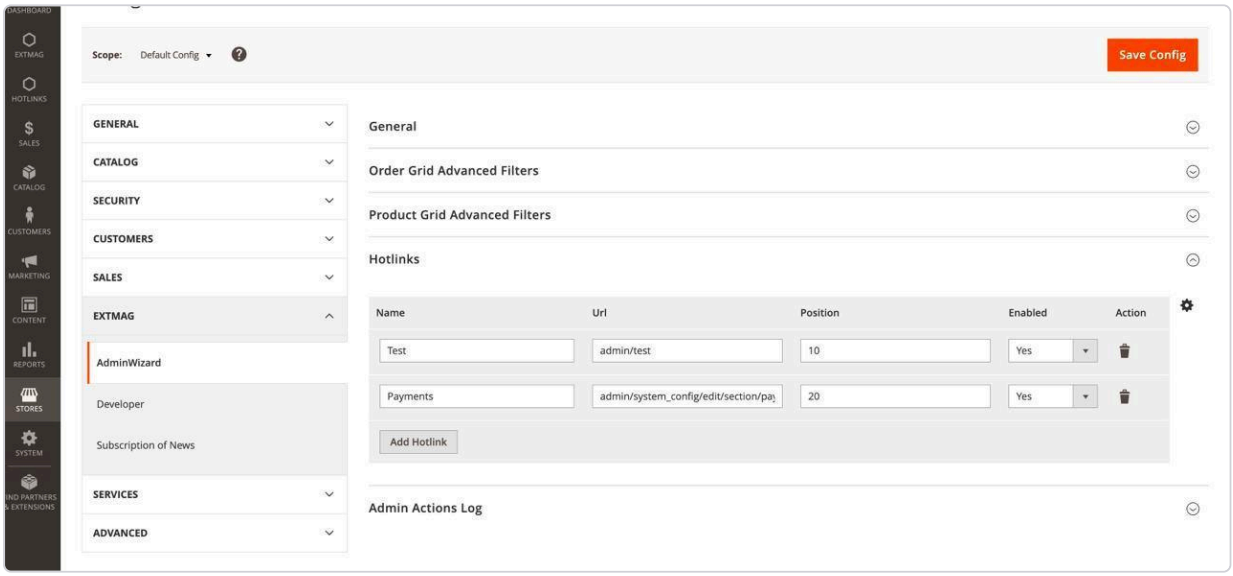
4 Hotlinks

Quick-access shortcuts to admin pages you visit often. Two modes coexist and both render on the standard admin dashboard.

4.1 Manual hotlinks

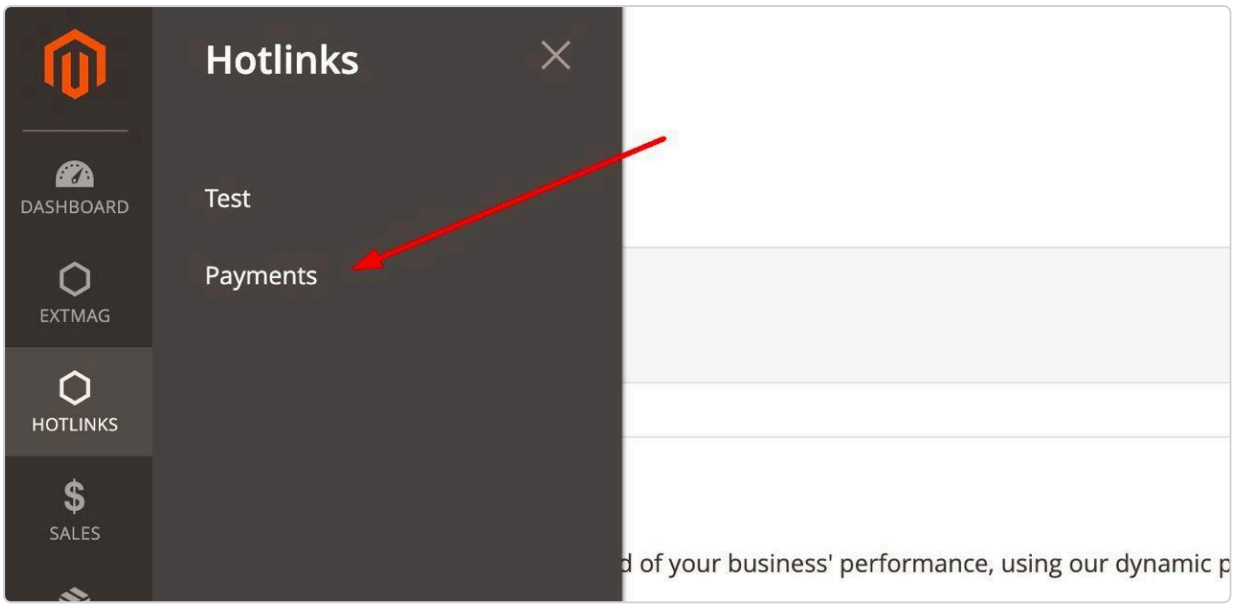
1. `Stores → Configuration → Extmag → AdminWizard → Hotlinks → Manual Hotlinks`.
2. *Add* a row. Enter Title and URL (full admin URL or relative path).
3. Save. Flush the configuration cache.
4. Refresh the admin dashboard — the hotlinks card lists every manual entry.

Manual hotlinks are global across all admin users and stored in `core_config_data` (serialized).



Manual Hotlinks configuration grid

The saved hotlinks render as a dedicated *Hotlinks* item in the admin sidebar, expanding into a panel that lists every entry.



Hotlinks panel in the admin sidebar

4.2 Auto hotlinks

1. `Stores → Configuration → Extmag → AdminWizard → Hotlinks → Auto-generate Hotlinks → Yes`.
2. Set *Auto Hotlinks Count* — the cap per user. Default 5.
3. Save and flush cache.

From now on, every admin controller dispatch increments a per-user `(url_path, visit_count)` counter in `extmag_admin_hotlink_stats`. The top N most visited pages per user surface on the dashboard hotlinks card. Counters update in real time on every page load.

5 Advanced Filters for sales and product grids

These filters allow rule-based search on any product or sales attribute, including fields that Magento's native grid filter row does not expose (item-level conditions, address fields, customer attributes).

5.1 Coverage

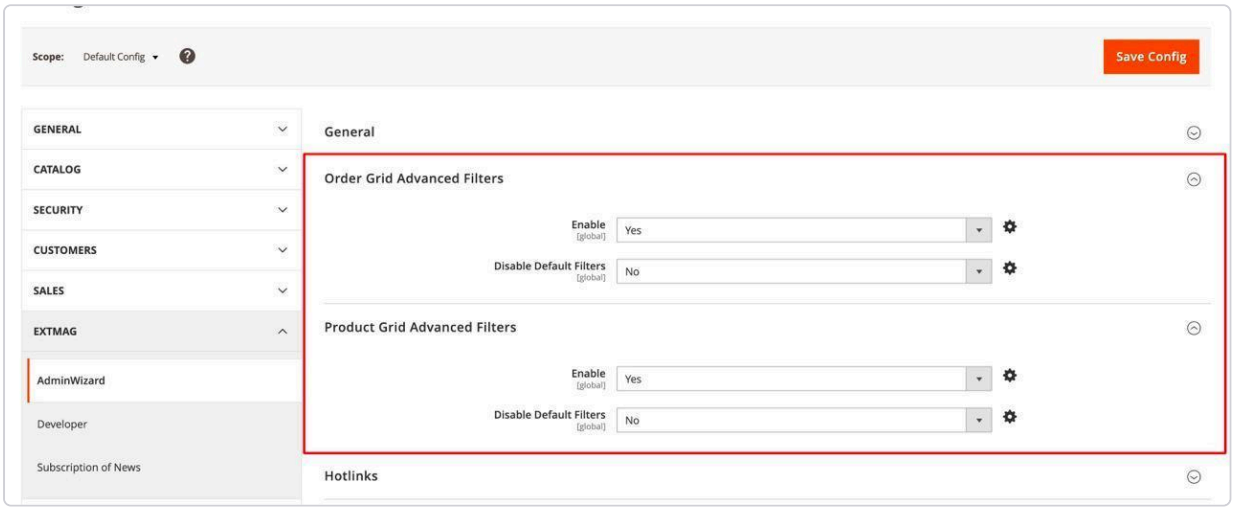
Five sales / catalog grids:

- `Sales → Orders`
- `Sales → Invoices`

- `Sales → Shipments`
- `Sales → Credit Memos`
- `Catalog → Products`

5.2 Enabling

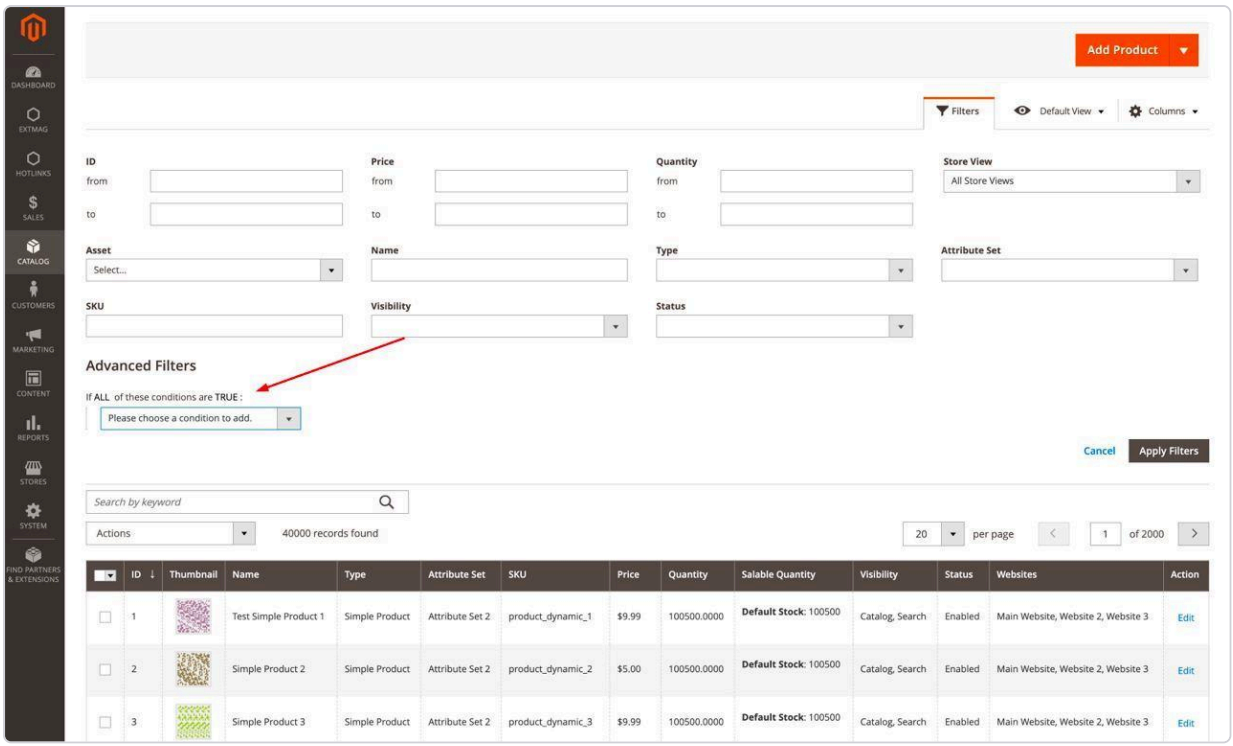
1. `Stores → Configuration → Extmag → AdminWizard`.
2. Open the *Advanced Filters* group for the grid you want — Order, Invoice, Shipment, Creditmemo, or Product.
3. *Enable = Yes*.
4. *Disable Default Filters = Yes* to hide Magento's native filter row if you find it confusing alongside the advanced filter. Default is *No* — both filter rows coexist.
5. Save and flush cache.



Per-grid Advanced Filters toggle

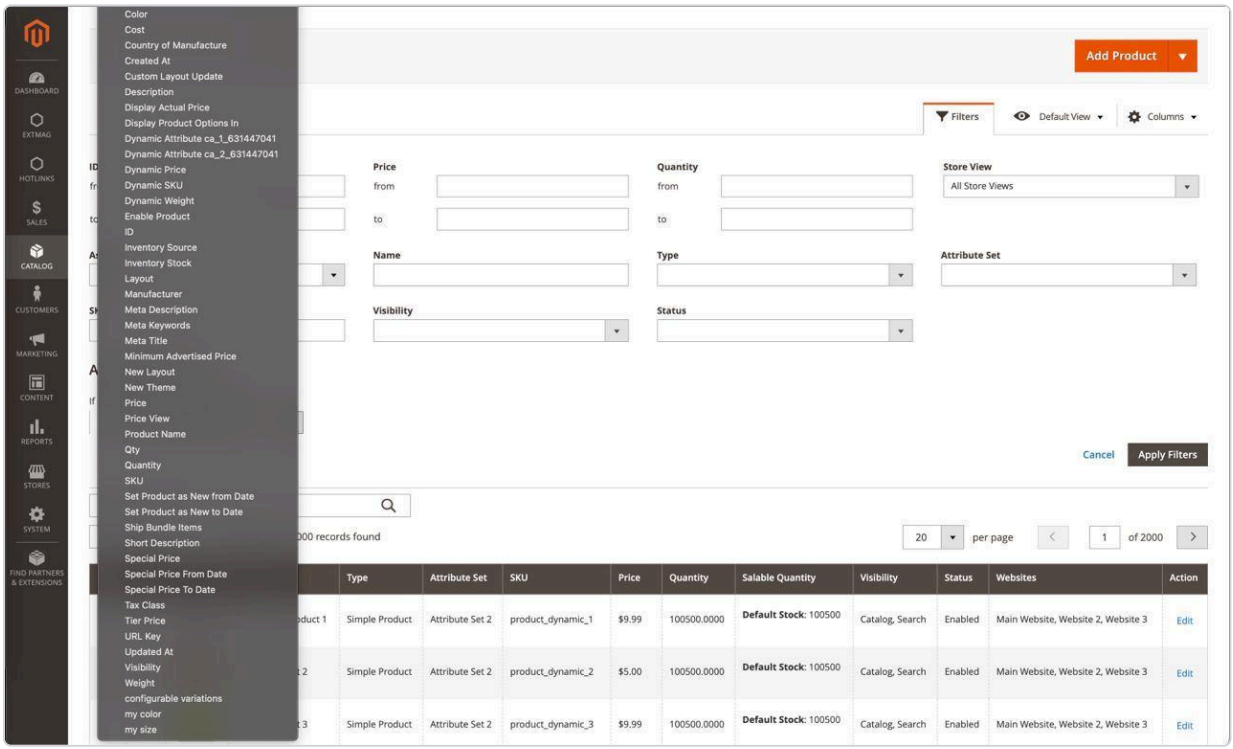
5.3 Using

1. Open the affected grid (e.g. `Sales → Orders`).
2. Click the *Filters* button at the top right.
3. Find the *Advanced Filters* fieldset above the standard filter row.
4. Use the conditions tree (the same UI as the Cart Price Rules conditions) to compose your filter — AND / OR groups, any attribute.
5. *Apply Filters*. The grid refreshes against the composed query.



Advanced Filters fieldset on the Products grid

Picking a condition opens the attribute selector — every product attribute, EAV or static, is available in one alphabetised list.

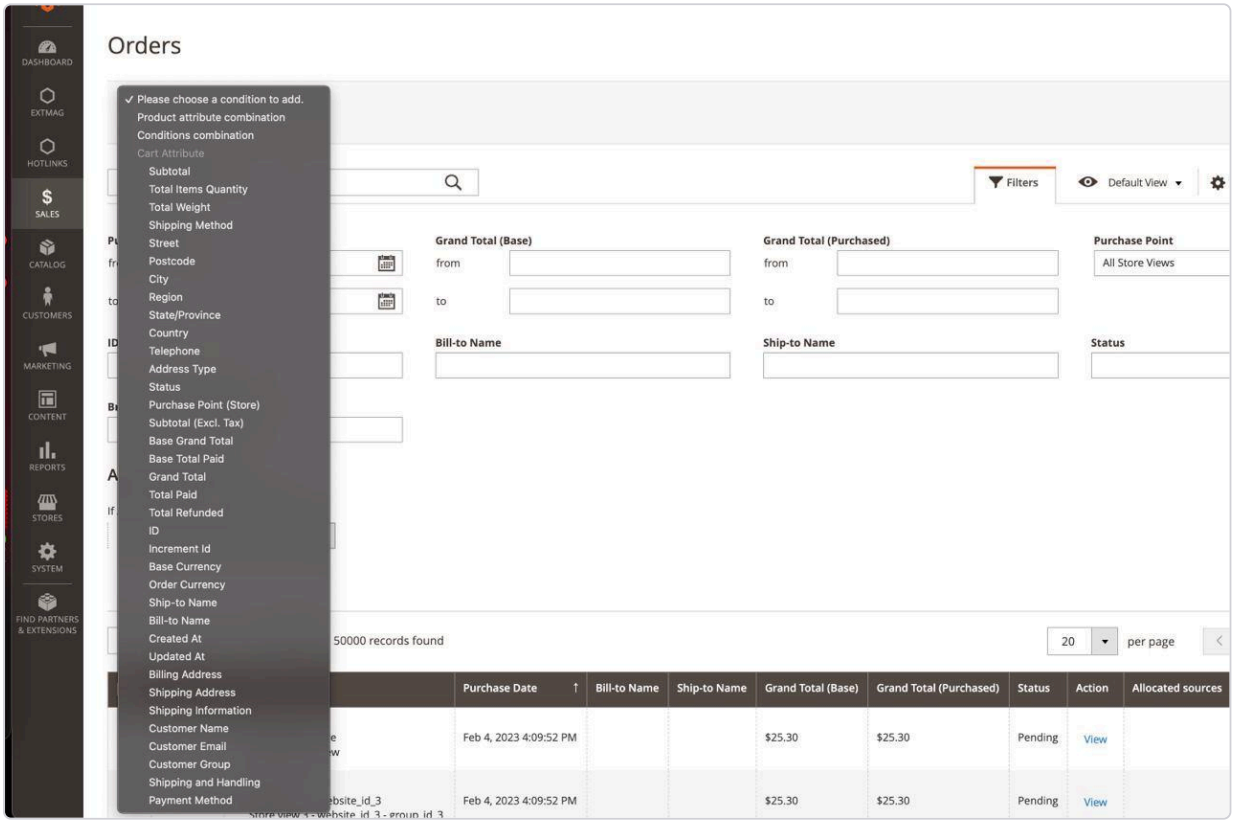


Attribute selector for product Advanced Filters

A finished condition reads back as plain English. The composed query is applied with one click.

Single condition: Price equals or less than 5

The Orders grid uses the same UI but pulls in sales-specific attributes — subtotal, shipping method, billing / shipping address fields, customer attributes, line-item conditions.



Order grid Advanced Filters attribute list

The conditions tree saves to the URL bookmark — share the URL to share the filter, or save it via Magento's grid bookmark feature.

5.4 Reset

Click the *Clear* button (next to *Apply Filters*) to drop every advanced condition and re-apply.

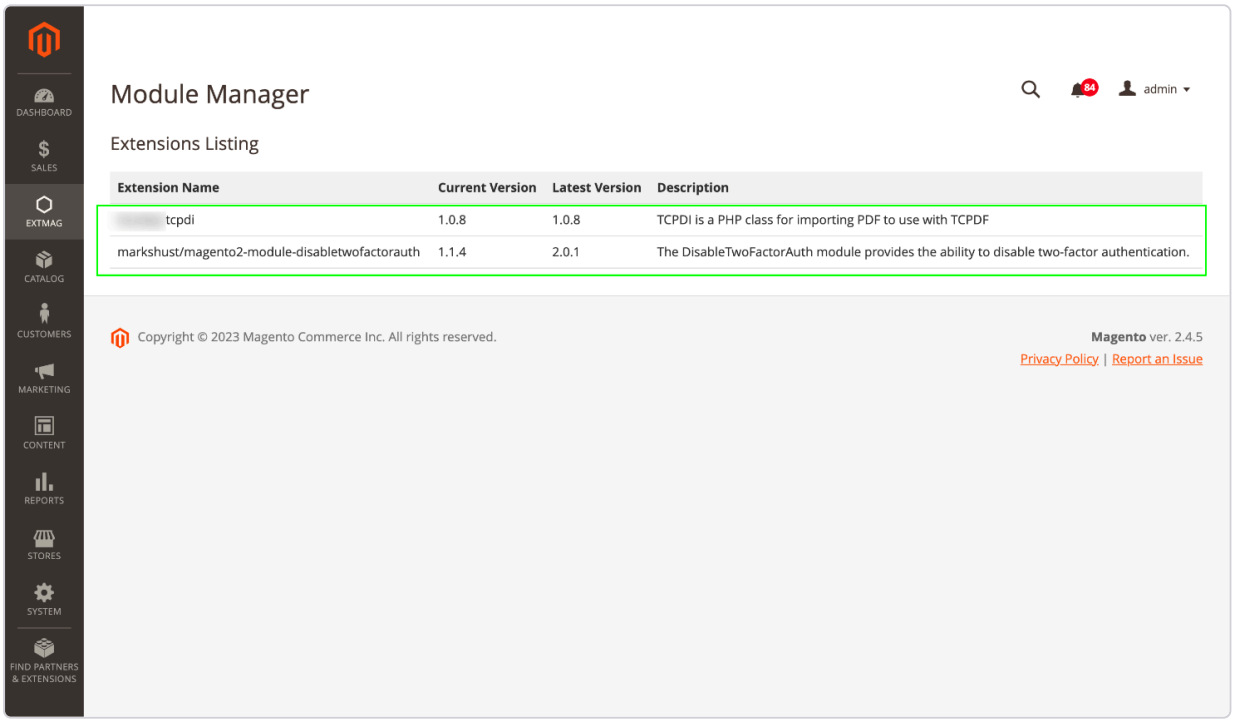
6 Extensions List

`Extmag → AdminWizard → Extensions List`.

The grid lists every installed Magento module with:

- Module name and namespace.
- Composer package, current installed version.
- Latest available version (where the module advertises an update feed).
- Enabled / disabled status.
- Origin (vendor) — Magento-vendored modules are deprioritised in the default sort.

This is the page to consult before a Magento upgrade or before raising a ticket with a vendor — version mismatches across the install stack are the most common root cause of obscure admin errors.



Module Manager — Extensions Listing

7 Environment Info

`Extmag → AdminWizard → Environment Info` collapses every diagnostic the support desk usually asks for into a single screen:

- PHP version, `memory_limit`, `max_execution_time`, loaded extensions.
- Magento version, edition, current mode (`developer` / `production` / `default`).

- Database engine, version, total size, table count.
- Disk space for `var/log`, `var/cache`, `pub/media`, `pub/static`, `generated`.
- Redis / Varnish / Elasticsearch reachability and version (where configured).
- List of every installed module with its declared version.

Use this page when filing a vendor ticket — the section selection mirrors what most extension support teams ask for.

8 Plugin & Preference Viewer

`Extmag → AdminWizard → Plugin & Preference Viewer`.

Two sub-grids:

- **Plugins** — every active plugin: target class, target method, plugin type (`before` / `after` / `around`), plugin class, declaring module.
- **Preferences** — every preference: interface, concrete class, declaring module.

Conflict detection highlights cases where two or more plugins target the same method (the *Plugin Conflicts* sub-grid). Multi-`before` and multi-`after` plugins are not necessarily a bug — the grid only surfaces them so you can reason about them.

Search across class, method and module to track down the source of unexpected behaviour during a third-party module triage.

9 Cron Monitor

`Extmag → AdminWizard → Cron Monitor` consolidates every cron job into one grid:

Column	Source
Job code	Magento cron registry.
Group	Magento cron registry.
Last status	`cron_schedule.status` for the most recent row.
Last run	`cron_schedule.executed_at` for the most recent row.
Duration	`executed_at − started_at`.
Next scheduled	`cron_schedule.scheduled_at` for the soonest `pending` row.

Per-row actions:

- **Run Now** — inserts a `pending` row in `cron_schedule` for the current minute. The Magento default cron group picks it up on the next tick.
- **History** — opens the full `cron_schedule` log scoped to that job.

The history grid backs onto Magento's own `cron_schedule` table, so filters and sorts work against indexed columns.

10 Config Diff

`Extmag → AdminWizard → Config Diff` lists every configuration value that has been overridden at website or store scope vs the module default.

Column	Notes
Path	`section/group/field` path.
Default value	From the module's `etc/config.xml` (or empty if not declared).
Scope	`default`, `website (<code>)`, or `store (<code>)`.
Override value	Current value at that scope.

Filter and search work across path and value. The *Export CSV* action dumps the visible filter result for offline review or audit.

11 Admin Sessions

`Extmag → AdminWizard → Admin Sessions` lists every row in `admin_user_session` (Magento Security module).

Column	Notes
Admin user	Username and email.
IP	Client IP recorded by Magento_Security at session start.
Status	`active` / `logged_out` / `expired`.
Created at	Session start.
Last activity	Most recent admin request.

Per-row action **Force Logout** sets the session row to `logged_out` — the next request from that browser is forced through the login screen. Force Logout requires the `Extmag_Improver::admin_sessions_force_logout` ACL resource on the operator's role.

Magento does not store admin user-agent, so the grid does not expose one.

12 Indexer Manager

The Indexer Manager surfaces from two places:

- `System → Tools → Index Management` — the native grid, enriched with `backlog` and `last_error` columns.
- `Extmag → AdminWizard → Indexer Manager → <indexer>` (linked from the per-row View action) — full detail page.

12.1 Detail view

Section	Contents
Overview	Title, status, mode, last executed, backlog count.
Run history	Last N rows from `extmag_indexer_history` — manual / cron / other.
Dependencies	Direct dependencies and dependents with clickable links to those indexers.
Control	Pause toggle and Throttle (minutes) — see below.

12.2 Reindex action

The detail page exposes:

- **Reindex** — runs `reindexAll()` on this one indexer.
- **Set Mode** — toggle between *Update on Save* and *Update on Schedule*.
- **Mass Reindex** from the native grid — reindex multiple indexers in one go.

Each call is logged into `extmag_indexer_history` with status, triggering user, duration, and (on failure) the error message.

12.3 Pause and Throttle

The control gate is implemented by a plugin on `Magento\Framework\Mview\View::update`.

- **Pause** — the mview update is skipped entirely. The indexer status stays *valid* until upstream data changes, then becomes *outdated* and stays there until pause is lifted. No data loss; the changelog table keeps growing.
- **Throttle (minutes)** — the mview update runs at most once per N minutes. Useful on indexers that recompute on every save event but whose downstream views can tolerate a few minutes of staleness.

Paused or throttled rows still appear in the Magento native grid with their normal status — only the mview update is gated. Manual *Reindex* from the detail view always runs, bypassing the gate.

12.4 When to pause / throttle

Common cases:

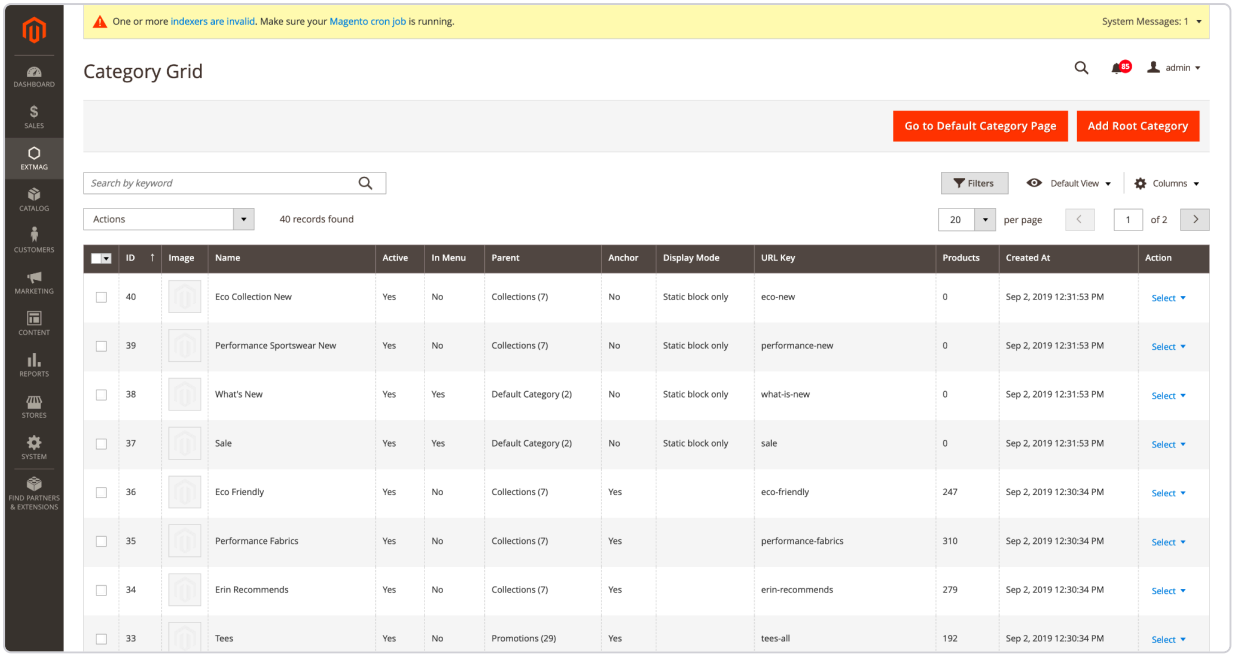
- Bulk product imports — pause `catalog_category_product` and `catalogsearch_fulltext` during the import, reindex once at the end.
- Long-running data migrations — pause every indexer except `stock` during the migration window.
- Memory-pressured environment — throttle high-traffic indexers (e.g. `catalogrule_product`) to a few minutes between passes.

Lift pause / throttle as soon as the operation completes. A long-paused indexer accumulates a large changelog backlog and the eventual catch-up pass will be expensive.

13 Categories Grid

`Catalog → Inventory → Categories Grid` replaces the standard tree view with a flat sortable, filterable grid.

Column	Notes
ID	Category entity id.
Name	Editable inline on the per-row edit form.
Path	Slash-separated category path.
Level	Tree level.
Is active	Yes / No toggle, mass actionable.
Position	
Product count	Live count from the category index.
Updated at	



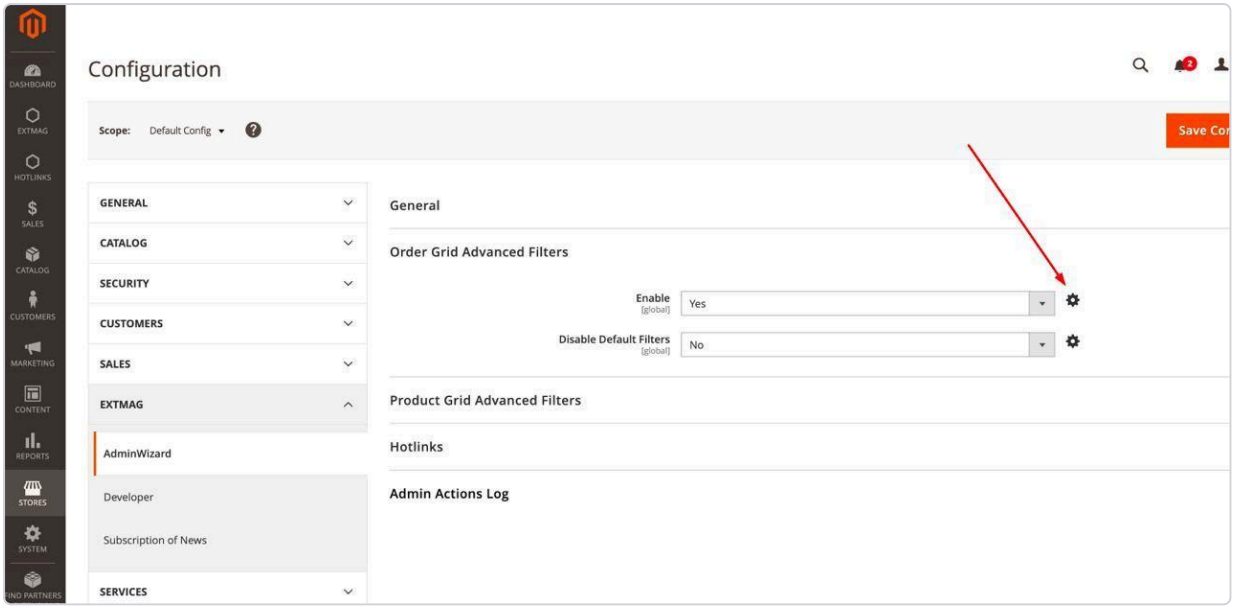
Flat Categories Grid replacing the tree view

Useful operations:

- Filter by *Is active = No* + mass action *Enable* to re-publish a batch of categories that were disabled by mistake.
- Sort by *Product count = 0* to find empty categories for cleanup.
- Per-row edit opens the standard category edit page.

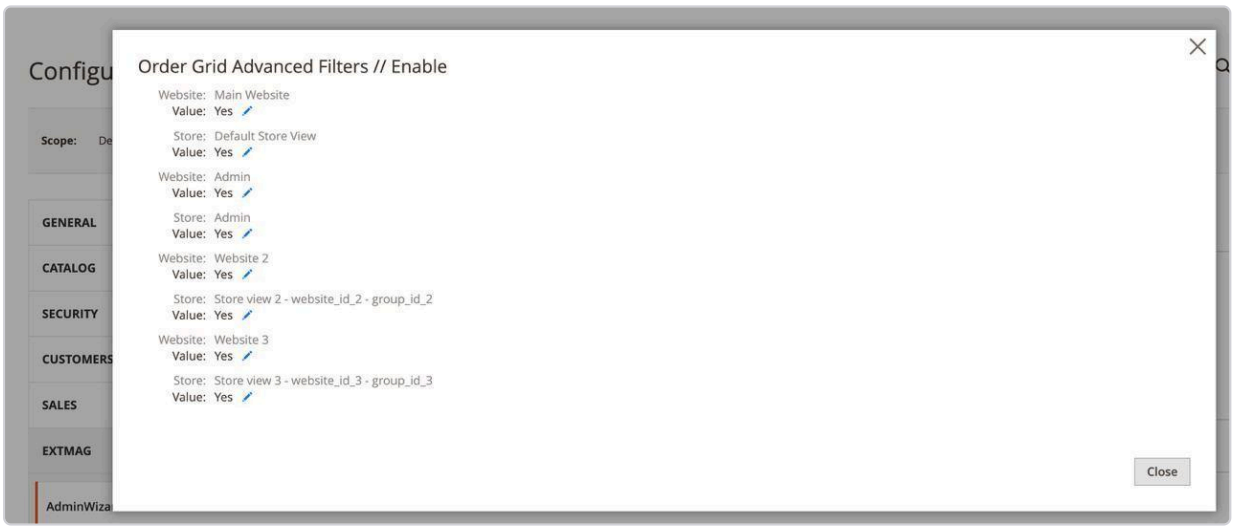
14 Multistore Configuration

Inside `Stores → Configuration → ...` any AdminWizard-managed field shows a small **gear** icon next to the input.



Gear icon next to the Advanced Filters Enable field

Click the gear to open a popover that lists the value of this exact configuration field on every store in the install — at a glance, no scope switch required.



Per-store value popover for the Order Grid Advanced Filters toggle

This is especially useful when verifying that the same Pricer Set, shipping rule, or tax rule resolves to the same value across all websites and stores after a configuration import.

15 Save DB tables on module removal

When the *Save DB tables* feature is on (default), the schema diff plugin suppresses `DROP TABLE` statements emitted by `setup:upgrade` for tables declared by disabled modules. This prevents the common foot-gun where a developer disables a module temporarily and Magento drops its data on the next deploy.

The feature is gated by `extmag_improver/general/keep_db_tables = 1`. Set to `0` to fall back to Magento's default behaviour.

Tables stay even after the module is removed from `composer.lock`. To actually drop them, run the `DROP TABLE` statements manually — the Installation Guide lists them.

16 Two Factor Auth bypass

`Stores → Configuration → Extmag → AdminWizard → Two Factor Auth` exposes three switches:

Field	Effect
Enable 2FA	Yes = bypass off (Magento_TwoFactorAuth runs as upstream). No = bypass on globally.
Enable 2FA for API Token Generation	Same toggle scoped to the admin API token endpoint.
Disable 2FA in Developer Mode	Yes = the bypass is itself disabled in developer mode (the bypass is on in prod only).

Use the bypass when running a load test, when an upstream identity provider already enforces MFA at the perimeter, or when restoring a test environment from a production backup that has 2FA secrets bound to the production hostname.

Do not use the bypass on a production environment exposed to the public internet without compensating controls. Magento_TwoFactorAuth is a security control — switching it off shifts the burden onto your perimeter.

17 Logs Manager Pro and Backup Manager Pro

Both are bundled at no extra cost via the `extmag/logs-core` and `extmag/backup-core` composer dependencies. Their features and menus remain unchanged from their respective module documentation — the only difference under AdminWizard is that the licence is included.

18 Troubleshooting

18.1 Hotlinks card missing on the dashboard

- For manual hotlinks: confirm at least one row in *Stores → Configuration → Extmag → AdminWizard → Hotlinks → Manual Hotlinks* and a cache flush after saving.
- For auto hotlinks: confirm *Auto-generate Hotlinks = Yes* and that you have visited at least one admin page since enabling it.

18.2 Advanced filter fieldset missing on a grid

- Check `Stores → Configuration → Extmag → AdminWizard → <grid> Advanced Filters → Enable = Yes`.
- If you flipped that during the current admin session, log out and back in — the grid layout is cached in the admin session.
- `setup:di:compile` after enabling — the four per-entity `virtualType` definitions need to be in the compiled DI snapshot.

18.3 *Force Logout* button does not appear on Admin Sessions

Operator role lacks the `Extmag_Improver::admin_sessions_force_logout` ACL resource. Grant it on the role's permissions tab.

18.4 Indexer status stays *outdated* even after pause is lifted

Paused indexers stop applying mview updates. The changelog table fills up while paused; the first run after pause-off processes the whole backlog. Wait for the next tick or click *Reindex* in the detail view to force a synchronous catch-up.

18.5 Categories Grid says *Class "…" does not exist*

`bin/magento setup:di:compile`. The grid extends a Magento parent collection whose constructor fallback resolves a connection through DI; the DI snapshot must exist for the bind to succeed.

18.6 Plugin Viewer shows duplicate `before` plugins on one method

That is not necessarily a conflict — `before` plugins compose order-independent unless one of them modifies the arguments. The *Conflicts* sub-grid lists the actual conflicts (multiple `around` plugins, or `after` plugins whose return values overlap). Investigate those first.

18.7 Cron Monitor *Run Now* does nothing

The button enqueues a `pending` row in `cron_schedule`. The Magento default cron group must be ticking for the row to fire. Verify with `bin/magento cron:status` and `crontab -l | grep magento`. The AdminWizard module does not own a cron daemon — it only triggers Magento's.

18.8 Extensions List shows no *Latest version*

Some vendors do not publish an update feed for their packages. The grid displays the installed version only in that case. Composer's `composer outdated` is the authoritative source when the column is empty.

18.9 2FA bypass is on, but admin still sees the OTP prompt

`Disable 2FA in Developer Mode = Yes` (the default) means the bypass is itself disabled in developer mode — exactly so a developer cannot accidentally lock the production behaviour out of their local env. Flip to *No* if you genuinely want the bypass active in developer mode too.

18.10 DB tables disappeared after disabling a module

- Confirm `keep_db_tables = 1` in *Stores → Configuration → Extmag → AdminWizard → General*.
- If it was *0* at the time of `setup:upgrade`, the tables were dropped by Magento's normal schema diff. Restore from backup — the AdminWizard feature only prevents future drops, it does not recover past ones.