

# Reference Manual

*AdminWizard for Magento 2.x — Extmag\_Improver*

2026-05-20

- 1 Module metadata
- 2 Admin routes
- 3 ACL resource tree
- 4 System configuration reference
  - 4.1 General ( `extmag_improver/general/...` )
  - 4.2 Order / Invoice / Shipment / Creditmemo / Product Advanced Filters
  - 4.3 Hotlinks ( `extmag_improver/hotlinks/...` )
  - 4.4 Two Factor Auth ( `extmag_improver/twofactorauth/...` )
- 5 Database schema
  - 5.1 `extmag_admin_hotlink_stats`
  - 5.2 `extmag_indexer_history`
  - 5.3 `extmag_indexer_control`
- 6 Plugin and observer wiring
  - 6.1 Sales-grid advanced filters
  - 6.2 Indexer Manager plugins
  - 6.3 2-Factor Auth bypass
  - 6.4 DB schema preservation
  - 6.5 Auto Hotlinks tracker
- 7 Indexer Manager — pause and throttle semantics
- 8 UI components
- 9 Frontend JavaScript and templates
- 10 Cron jobs
- 11 Performance and scale notes

# 1 Module metadata

Key	Value
Magento module name	Extmag_Improver
Composer package	extmag/improver
PSR-4 namespace	Extmag\Improver\
Required deps	extmag/core, extmag/logs-core, extmag/backup-core
Load sequence after	Magento_Backend, Extmag_Core, Magento_SalesRule
Admin route frontName	improver
Admin menu root	Extmag → AdminWizard (parent: Extmag_Core::main_menu)

There is no `webapi.xml` and no public PHP service contract. AdminWizard is an internal admin toolbox — every feature is reached through the admin UI, system configuration, or per-feature `etc/di.xml` plugins.

# 2 Admin routes

Single backend router `admin` with one route:

Route id	Module	Front name
improver	Extmag_Improver	improver

URL pattern: `<admin>/improver/<controller>/<action>`. Examples below list the menu item, ACL resource, and the controller folder under `Controller/Adminhtml/`.

Menu item	URL	ACL resource	Controller folder
Dashboard	improver/dashboard/index	Extmag_Improver::improver	Dashboard/
Extensions List	improver/extension/index	Extmag_Improver::extensions_manager	Extension/
Environment Info	improver/environment/index	Extmag_Improver::environment_info	Environment/
Plugin & Preference Viewer	improver/diViewer/index	Extmag_Improver::di_viewer	DiViewer/
Cron Monitor	improver/cron/index	Extmag_Improver::cron_monitor	Cron/
Config Diff	improver/configDiff/index	Extmag_Improver::config_diff	ConfigDiff/
Admin Sessions	improver/adminSession/index	Extmag_Improver::admin_sessions	AdminSession/
Categories Grid	improver/category/grid	Magento_Catalog::categories	Category/
Indexer Manager (detail view)	improver/indexer/view	Extmag_Improver::indexer_manager	Indexer/
Configuration	system_config/edit/section/extmag_improver	Extmag_Improver::improver_config	(Magento native)

# 3 ACL resource tree

```

Magento_Backend::admin
├── Extmag_Improver::improver          AdminWizard root
│   ├── Extmag_Improver::extensions_manager  Extensions List
│   ├── Extmag_Improver::environment_info    Environment Info
│   ├── Extmag_Improver::di_viewer           Plugin & Preference Viewer
│   ├── Extmag_Improver::cron_monitor        Cron Monitor
│   ├── Extmag_Improver::config_diff         Config Diff
│   ├── Extmag_Improver::admin_sessions      Admin Sessions
│   │   ├── Extmag_Improver::admin_sessions_force_logout  Force Logout
│   ├── Extmag_Improver::indexer_manager     Indexer Detail View
│   │   ├── Extmag_Improver::indexer_reindex  Reindex Data (mass action)
│   │   └── Extmag_Improver::indexer_control  Pause and throttle indexers
│   └── Extmag_Improver::hotlinks            Hotlinks

```

A separate branch under `Magento_Backend::stores` → `stores_settings` → `Magento_Config::config` → `Extmag_Core::config` → `Extmag_Improver::config` gates the System Configuration section.

## 4 System configuration reference

Stores → Configuration → Extmag → AdminWizard ( `section_id = extmag_improver` ).

### 4.1 General ( `extmag_improver/general/...` )

Field	Default	Notes
<code>enable</code>	Yes	Master switch for the AdminWizard menu items.
<code>keep_db_tables</code>	Yes	Preserve disabled-module tables (see <i>Save DB tables</i> feature).

### 4.2 Order / Invoice / Shipment / Creditmemo / Product Advanced Filters

Five identical groups, identifiers `order_advanced_filter`, `invoice_advanced_filter`, `shipment_advanced_filter`, `creditmemo_advanced_filter`, `product_advanced_filter`.

Field	Default	Source model
<code>enable</code>	Yes	<code>Magento\Config\Model\Config\Source\Yesno</code>
<code>disable_default_filters</code>	No	<code>Magento\Config\Model\Config\Source\Yesno</code>

`disable_default_filters = Yes` hides the native Magento filter row so operators do not accidentally mix the two filter mechanisms.

### 4.3 Hotlinks ( `extmag_improver/hotlinks/...` )

Field	Default	Notes
<code>auto_hotlinks_enable</code>	No	Track admin page visits per user and surface top N as auto hotlinks.
<code>auto_hotlinks_count</code>	5	Cap for auto-generated hotlinks per user. Validate-digits.
<code>hotlinks_field</code>	empty	Manual hotlinks array (serialized) — title + URL pairs.

### 4.4 Two Factor Auth ( `extmag_improver/twofactorauth/...` )

Field	Default	Notes
<code>enable</code>	Yes	Master switch. No means the bypass plugins return early everywhere.
<code>enable_for_api_token_generation</code>	Yes	Apply 2FA to the admin API token endpoint specifically.
<code>disable_in_developer_mode</code>	Yes	When Magento is in developer mode, the bypass is itself disabled.

## 5 Database schema

Declared in `etc/db_schema.xml`. All three tables use the `default` resource and `innodb` engine.

### 5.1 `extmag_admin_hotlink_stats`

Column	Type	Notes
<code>stat_id</code>	int unsigned, AI	Primary key.
<code>user_id</code>	int unsigned	Admin user id. Part of the unique key with <code>url_path</code> .
<code>url_path</code>	varchar(255)	Controller path ( <code>module/controller/action</code> ).
<code>title</code>	varchar(255)	Page title snapshot.
<code>visit_count</code>	int unsigned, default 0	
<code>last_visited</code>	timestamp, on update CURRENT	

Indexes: unique ( `user_id`, `url_path` ), btree ( `user_id`, `visit_count` ).

### 5.2 `extmag_indexer_history`

Column	Type	Notes
<code>history_id</code>	int unsigned, AI	Primary key.
<code>indexer_id</code>	varchar(128)	Magento indexer id.
<code>status</code>	varchar(20)	success or error.
<code>triggered_by</code>	varchar(20)	manual, cron, other.
<code>admin_user_id</code>	int unsigned, nullable	Admin user for manual runs.
<code>started_at</code>	timestamp	Defaults to <code>CURRENT_TIMESTAMP</code> .
<code>duration</code>	float, nullable	Seconds.
<code>message</code>	text, nullable	Error message or info.

Indexes: btree ( `indexer_id`, `started_at` ), btree ( `started_at` ).

### 5.3 `extmag_indexer_control`

Column	Type	Notes
<code>indexer_id</code>	varchar(128)	Primary key. Magento indexer id.
<code>throttle_minutes</code>	int unsigned, nullable	Minimum interval between mview updates. <code>null / 0</code> = off.
<code>paused</code>	smallint unsigned, default 0	1 = mview updates skipped entirely.
<code>last_processed_at</code>	timestamp, nullable	Last successful mview update start.
<code>updated_by_user_id</code>	int unsigned, nullable	Admin user who last changed control.
<code>updated_at</code>	timestamp, on update CURRENT	

## 6 Plugin and observer wiring

Configured in `etc/di.xml`.

### 6.1 Sales-grid advanced filters

For each of `Order`, `Invoice`, `Shipment`, `Creditmemo`, the module ships four `virtualType` entries that compose the `Magento_SalesRule` condition combinator into a per-entity advanced filter:

Pattern	Purpose
<code>Extmag\Improver\Model\Rule\Condition\&lt;Entity&gt;\CombineFactory</code>	Factory producing the per-entity Combine root.
<code>Extmag\Improver\Model\Rules\&lt;Entity&gt;</code>	Rule model carrying the entity code.
<code>Extmag\Improver\Ui\DataProvider\AdvancedFilters\&lt;Entity&gt;\DataProvider</code>	UI Component data provider injected into the grid.

Pattern	Purpose
Extmag\Improvev\Block\Adminhtml\Rules\<Entity>\Conditions	Block rendering the conditions tree in the filter.

The same four `virtualType` block is duplicated four times in `di.xml` — extending to a fifth sales entity is a copy-rename of the block.

The Product grid uses `Magento\Catalog\Ui\DataProvider\Product\ProductDataProvider` directly, with a `BeforeProductGetData` plugin in `Extmag\Improvev\Plugin\Ui\DataProvider\Product\` to inject the advanced filter selection before the parent runs.

## 6.2 Indexer Manager plugins

Magento type	Plugin class	Behaviour
Magento\Indexer\Model\Indexer	Extmag\Improvev\Plugin\IndexerHistoryPlugin	After <code>reindex</code> into <code>extmag_in</code>
Magento\Indexer\Ui\DataProvider\Indexer\DataCollection	Extmag\Improvev\Plugin\Indexer\DataCollectionPlugin	Enriches the grid with <code>ba</code> <code>last_erro</code>
Magento\Framework\Mview\View	Extmag\Improvev\Plugin\Indexer\MviewUpdatePlugin	Around <code>upd</code> <code>extmag_in</code> and pauses <code>,</code> updates.

## 6.3 2-Factor Auth bypass

Magento type	Plugin class	E
Magento\TwoFactorAuth\Model\TfaSession	Extmag\Improvev\Plugin\BypassTwoFactorAuth	V i =
Magento\TwoFactorAuth\Model\AdminAccessTokenService	Extmag\Improvev\Plugin\BypassTwoFactorAuthForApiTokenGeneration	S

## 6.4 DB schema preservation

Plugin `Extmag\Improvev\Plugin\MagentoFramework\Setup\Declaration\Schema\Diff\DiffManager` filters the schema diff so tables owned by disabled modules are not issued a `DROP TABLE`. Gated by `extmag_improvev/general/keep_db_tables`.

## 6.5 Auto Hotlinks tracker

Plugin `Extmag\Improvev\Plugin\AdminPageTracker` (wired through `Plugin/Controller/Adminhtml/...`) increments `extmag_admin_hotlink_stats.visit_count` on every backend controller dispatch where an admin user is logged in.

# 7 Indexer Manager — pause and throttle semantics

The `mview` update plugin reads `extmag_indexer_control` once per call:

```

if paused == 1                -> skip update, return
if throttle_minutes > 0
    and now() - last_processed_at < throttle    -> skip update, return
    otherwise                                    -> proceed, update last_processed_at

```

`last_processed_at` is written only on a successful pass. A failing update leaves the old timestamp in place so the next tick re-attempts without waiting an extra throttle window.

Paused indexers also stop logging into `extmag_indexer_history` — the plugin gates before the indexer runs.

## 8 UI components

`view/adminhtml/ui_component/` ships listing components for each grid:

Component name	Backing collection / source
<code>improver_category_grid_listing</code>	<code>Extmag\Improver\Model\ResourceModel\Category\Grid\Collection</code>
<code>improver_cron_history_listing</code>	virtualType on Magento <code>cron_schedule</code> table
<code>improver_admin_session_listing</code>	type on Magento <code>admin_user_session</code> table
<code>improver_extension_listing</code>	Composer + module list collection
<code>improver_plugin_listing</code>	DI graph walk
<code>improver_preference_listing</code>	DI graph walk
<code>improver_plugin_conflicts_listing</code>	DI graph walk, conflicts subset
<code>improver_preference_overrides_listing</code>	DI graph walk, overrides subset
<code>improver_config_diff_listing</code>	<code>core_config_data</code> vs <code>etc/config.xml</code> defaults
<code>improver_cron_listing</code>	Magento cron registry walk
<code>sales_order_grid /</code> <code>sales_order_invoice_grid /</code> <code>sales_order_shipment_grid /</code> <code>sales_order_creditmemo_grid /</code> <code>product_listing</code>	Overrides to inject the advanced filter slot.
<code>category_form / new_category_form</code>	Category Grid edit form.

## 9 Frontend JavaScript and templates

Custom admin JS for the sales-grid advanced filters lives in `view/adminhtml/web/js/grid/filters/filters.js`; the matching template in `view/adminhtml/templates/promo/fieldset.phtml` (the `Promo` namespace is reused because the underlying conditions tree comes from `Magento_SalesRule`). Both files mark their delta from upstream with `/* Extmag Changes */ / /* END Extmag Changes */` blocks.

## 10 Cron jobs

The module ships no `crontab.xml`. Cron Monitor reads from Magento's `cron_schedule` table and the cron registry; it does not register jobs of its own.

The two background concerns the module owns are:

- Hotlink stats — updated synchronously on each admin controller dispatch by the `AdminPageTracker` plugin. No deferred work.
- Indexer history — appended synchronously after each `reindexAll / reindexList` invocation by the `IndexerHistoryPlugin`. No deferred work.

## 11 Performance and scale notes

- The DI Viewer walks the compiled `di.xml` graph from `generated/`. After `setup:di:compile` the data is read from disk in tens of milliseconds even on a thousand-module install.
- The Cron Monitor grid uses an indexed scan on `cron_schedule(scheduled_at)` — paging stays responsive past hundreds of thousands of rows.
- The Indexer Manager `backlog` column reads each indexer's changelog table (`<view>_cl`) with `SELECT COUNT(*)`. On indexers with very long changelogs this can take a few hundred milliseconds per row, all in parallel inside one grid render. Trim changelogs with the standard Magento `mview` cleanup if grid render time becomes a concern.
- The Config Diff grid materialises only rows that differ from defaults — the full `core_config_data` scan happens once per render.