

Reference Manual

Extended Shipping Rates for Magento 2.x — Extmag_ExtendedRates

2026-05-22

- 1 Module overview
- 2 Carrier code and config path
- 3 ACL tree
- 4 System configuration reference
- 5 Service contracts (PHP API)
 - 5.1 `Extmag\ExtendedRates\Api\ShippingMethodsRepositoryInterface`
 - 5.2 `Extmag\ExtendedRates\Api\PriceRulesRepositoryInterface`
 - 5.3 `Extmag\ExtendedRates\Api\BlackoutRulesRepositoryInterface`
 - 5.4 `Extmag\ExtendedRates\Api\Data\ShippingMethodsInterface`
 - 5.5 `Extmag\ExtendedRates\Api\Data\PriceRulesInterface`
- 6 Database schema
 - 6.1 `extmag_extended_rates`
 - 6.2 `extmag_exrates_price_rules`
 - 6.3 `extmag_blackout_rules`
- 7 Carrier resolution pipeline
- 8 Schedule filter
 - 8.1 Frequencies
 - 8.2 `schedule_config` JSON shape
- 9 Blackout rule filter
- 10 Price calculation
 - 10.1 Price pattern syntax
- 11 Customer group resolution
- 12 CSV import / export
- 13 Adminhtml UI components
- 14 Hyva Theme integration
- 15 Known limitations

1 Module overview

`Extmag_ExtendedRates` is a Magento 2 shipping carrier module that extends the native shipping flow with three reusable, rule-driven entities:

| Entity | Table | Purpose |
|-----------------|---|---|
| Shipping Method | <code>extmag_extended_rates</code> | One offer at checkout. Conditions + base price + schedule + attached rules. |
| Price Rule | <code>extmag_exrates_price_rules</code> | Reusable price modifier with its own conditions and schedule. |
| Blackout Rule | <code>extmag_blackout_rules</code> | Reusable list of dates on which methods must be hidden. |

The carrier is registered under the code `extmagexrates` in `etc/config.xml`. At rate-request time the carrier filters its method collection by customer group, website, conditions, schedule and blackout rules, then optionally rewrites the price through the attached Price Rules.

2 Carrier code and config path

| Setting | Configuration path |
|----------------|--|
| Carrier code | <code>extmagexrates</code> |
| Active flag | <code>carriers/extmagexrates/active</code> |
| Carrier title | <code>carriers/extmagexrates/title</code> |
| Sort order | <code>carriers/extmagexrates/sort_order</code> |
| Debug flag | <code>carriers/extmagexrates/debug</code> |
| Specific error | <code>carriers/extmagexrates/specificerrmsg</code> |

`showmethod` is hard-coded to `1` in `etc/config.xml` — even when no matching method exists the carrier reports its presence rather than disappearing silently.

3 ACL tree

`Magento_Backend::admin` :

```

Extmag_ExtendedRates::methods          "Extended Shipping Rates"
  Extmag_ExtendedRates::methods_index  "View"
  Extmag_ExtendedRates::methods_save   "Create/Update"
  Extmag_ExtendedRates::methods_delete "Delete"
  Extmag_ExtendedRates::methods_export "Export"
  Extmag_ExtendedRates::methods_import "Import"
Extmag_ExtendedRates::rules            "Extended Price Rules"
  Extmag_ExtendedRates::rules_index    "View"
  Extmag_ExtendedRates::rules_save     "Create/Update"
  Extmag_ExtendedRates::rules_delete   "Delete"
Extmag_ExtendedRates::blackout_rules   "Blackout Rules"
  Extmag_ExtendedRates::blackout_rules_index "View"
  Extmag_ExtendedRates::blackout_rules_save "Create/Update"
  Extmag_ExtendedRates::blackout_rules_delete "Delete"

```

Assign roles at the leaf level for granular permissions — for example *View* without *Create/Update* gives an audit operator read-only access.

4 System configuration reference

Stores → Configuration → Sales → Shipping Methods → Extended Shipping Rates (carriers/extmagexrates).

| Field | Default | Source model / Type |
|------------|---------|--|
| active | No | Magento\Config\Model\Config\Source\Yesno |
| title | empty | text |
| sort_order | empty | text |
| export | n/a | button — exports current shipping methods to CSV |
| import | n/a | file — uploads CSV into media/extmag_sm_import |
| debug | No | Yesno |

Import is processed by `Extmag\ExtendedRates\Helper\ImportShippingMethods` (registered as the field's `backend_model`). Export is rendered by `Extmag\ExtendedRates\Block\Adminhtml\System\Config\ExportShippingMethods`.

5 Service contracts (PHP API)

The module ships PHP-only service contracts. There is no `webapi.xml`, so these APIs are not exposed over REST / SOAP — consume them via Magento DI from your own modules or scripts.

5.1 `Extmag\ExtendedRates\Api\ShippingMethodsRepositoryInterface`

CRUD for Shipping Methods.

```
public function save(ShippingMethodsInterface $method):  
    ShippingMethodsInterface;  
public function getById(int $entityId): ShippingMethodsInterface;  
public function getList(SearchCriteriaInterface $criteria):  
    ShippingMethodsSearchResultsInterface;  
public function delete(ShippingMethodsInterface $method): bool;  
public function deleteById(int $entityId): bool;
```

5.2 `Extmag\ExtendedRates\Api\PriceRulesRepositoryInterface`

CRUD for Price Rules. Same shape as the Shipping Methods repository, returning `PriceRulesInterface` instances.

5.3 `Extmag\ExtendedRates\Api\BlackoutRulesRepositoryInterface`

CRUD for Blackout Rules.

5.4 `Extmag\ExtendedRates\Api\Data\ShippingMethodsInterface`

Shipping Method entity accessors. Field constants exposed on the interface:

```
ENTITY_ID, ADMIN_TITLE, TITLE, CARRIER_TITLE, PRICE,  
CONDITIONS_SERIALIZED, STATUS, FREE_SHIPPING,  
CREATED_TIME, UPDATED_TIME, WEBSITE_IDS, CUSTOMER_GROUP_IDS,  
PRICE_RULE_IDS, DATE_FROM, DATE_TO, FREQUENCY, DAY_WEEK,  
AREA, BLACKOUT_RULE_IDS, SCHEDULE_CONFIG
```

Each constant has a paired getter / setter on the model (`getTitle` / `setTitle`, `getScheduleConfig` / `setScheduleConfig`, ...).

Comma-separated multiselect columns (`website_ids`, `customer_group_ids`, `price_rule_ids`, `blackout_rule_ids`) expose helper getters that return the value as an array (`getPriceRuleIdsAsArray`, `getBlackoutRuleIdsAsArray`, `getDayWeekAsArray`).

5.5 Extmag\ExtendedRates\Api\Data\PriceRulesInterface

Price Rule entity. Constants:

```
ENTITY_ID, TITLE, STATUS, PRICE, CONDITIONS_SERIALIZED,  
CREATED_TIME, UPDATED_TIME, DATE_FROM, DATE_TO,  
FREQUENCY, DAY_WEEK, SCHEDULE_CONFIG
```

6 Database schema

6.1 extmag_extended_rates

| Column | Type | Notes |
|-----------------------|----------------------|---|
| entity_id | int unsigned, AI | Primary key. |
| title | varchar(255) | Title shown at checkout. |
| admin_title | varchar(255), unique | Description shown in admin grid. Unique constraint. |
| carrier_title | varchar(255) | Per-method override for the carrier title. |
| status | int | 1 = active, 0 = inactive. |
| conditions_serialized | mediumtext | Magento Rule serialized condition tree. |
| free_shipping | int unsigned | 1 = honour native free shipping promotions. |
| website_ids | text | Comma-separated store / website ids. |
| customer_group_ids | text | Comma-separated customer group ids. |
| price | decimal(7,2) | Base shipping price. |
| price_rule_ids | mediumtext | Comma-separated Price Rule ids attached to this method. |
| frequency | varchar(50) | Legacy schedule frequency. Empty when schedule_config. |
| day_week | varchar(90) | Legacy day-of-week multiselect. |
| date_from | timestamp NULL | Legacy schedule start. |
| date_to | timestamp NULL | Legacy schedule end. |

| Column | Type | Notes |
|-------------------|---------------------------------------|---|
| created_time | timestamp DEFAULT CURRENT_TIMESTAMP | Insertion timestamp. |
| updated_time | timestamp ON UPDATE CURRENT_TIMESTAMP | Last update. |
| area | varchar(50) DEFAULT 'all' | all = Frontend + Admin; admin = Admin Only. |
| logo | text | JSON-encoded uploaded image metadata. |
| blackout_rule_ids | text NULL | Comma-separated non-global blackout rule ids. |
| schedule_config | mediumtext NULL | Multi-slot schedule JSON (replaces legacy columns). |

6.2 extmag_exrates_price_rules

| Column | Type | Notes |
|-----------------------|---------------------------------------|--|
| entity_id | int unsigned, AI | Primary key. |
| title | varchar(255), unique | Rule name. |
| status | int | 1 = active, 0 = inactive. |
| conditions_serialized | mediumtext | Magento Rule serialized condition tree. |
| price | varchar(50) | Price pattern (see below). |
| frequency | varchar(50) | Legacy schedule frequency. |
| day_week | varchar(90) | Legacy day-of-week multiselect. |
| date_from | timestamp NULL | Legacy schedule start. |
| date_to | timestamp NULL | Legacy schedule end. |
| created_time | timestamp DEFAULT CURRENT_TIMESTAMP | Insertion timestamp. |
| updated_time | timestamp ON UPDATE CURRENT_TIMESTAMP | Last update. |
| schedule_config | mediumtext NULL | Multi-slot schedule JSON. |
| blackout_rule_ids | text NULL | Comma-separated blackout rule ids attached to this rule. |

6.3 extmag_blackout_rules

| Column | Type | Notes |
|--------------|---------------------------------------|--|
| entity_id | int unsigned, AI | Primary key. |
| title | varchar(255), unique | Rule name. |
| status | tinyint unsigned | 1 = active, 0 = inactive. |
| is_global | tinyint unsigned | 1 = applies to every shipping method automatically. |
| dates_config | mediumtext NULL | JSON list of date ranges / single dates covered by the rule. |
| created_time | timestamp DEFAULT CURRENT_TIMESTAMP | Insertion timestamp. |
| updated_time | timestamp ON UPDATE CURRENT_TIMESTAMP | Last update. |

7 Carrier resolution pipeline

`Extmag\ExtendedRates\Model\Carrier::collectRates()` builds the rate list per Magento's standard carrier contract. The pipeline executes once per `RateRequest` :

1. Load every active shipping method whose:
 - `website_ids` contains the request store id,
 - `customer_group_ids` contains the cart customer's group id (admin orders read the group from `Backend\Model\Session\Quote`),
 - `area` matches the current Magento area — `admin` methods are hidden at the storefront.
2. For each candidate method, validate the cart against the method's serialized condition tree (`Magento\Rule\Model\AbstractModel`).
3. Apply the schedule filter — see [Schedule filter](#).
4. Apply the blackout rule filter — see [Blackout rule filter](#).
5. Compute the price — see [Price calculation](#).
6. Honour the native free shipping promotion when *Free Shipping = Yes* and the request flag is set.
7. Emit a `Magento\Quote\Model\Quote\Address\RateResult\Method` for each surviving method.

8 Schedule filter

The carrier reads `schedule_config` first; if empty it falls back to the legacy `frequency / date_from / date_to / day_week` columns. Both code paths share the same observable behaviour for single-slot schedules.

8.1 Frequencies

| Frequency | Triggers on |
|--|---|
| <code>` (empty) Always visible (frequency disabled). once Once during the [date_from, date_to] window, no yearly repeat. daily Every day, optionally within the chosen time-of-day slots and on the chosen weekdays. monthly On chosen days of the month, optionally within the chosen time-of-day slots. annually`</code> | On chosen month-day pairs every year, optionally within the chosen time-of-day slots. |

`weekly` is reserved in the source code but not surfaced in the *Application Frequency* dropdown.

8.2 `schedule_config` JSON shape

```

{
  "frequency": "monthly",
  "active_from": "2026-01-01",
  "active_to": "2026-12-31",
  "day_week": ["1", "2", "3", "4", "5"],
  "time_slots": [{"from": "09:00:00", "to": "18:00:00"}],
  "time_slots_monthly": [
    {"day": 1, "from": "00:00:00", "to": "23:59:59"},
    {"day_from": 15, "day_to": 20, "from": "09:00:00", "to": "18:00:00"}
  ],
  "time_slots_annually": [
    {"month_day": "12-24", "from": "00:00:00", "to": "23:59:59"},
    {"month_day_from": "06-01", "month_day_to": "08-31", "from": "09:00:00",
     "to": "18:00:00"}
  ]
}

```

Keys present depend on `frequency` :

- `daily` reads `time_slots[]` with `{from, to}`.
- `monthly` reads `time_slots_monthly[]` with either a single `day` or a `day_from / day_to` range, plus `{from, to}`.
- `annually` reads `time_slots_annually[]` with either a single `month_day` (MM-DD) or `month_day_from / month_day_to` range, plus `{from, to}`.

Multiple slots compose with OR — the method is visible if **any** slot matches.

9 Blackout rule filter

After the schedule allows a method, the carrier consults the `BlackoutRulesProvider` service. The provider loads every active blackout rule into a cached array (`CACHE_TAG = extmag_blackout_rules`, TTL 3600 s) and the carrier walks:

1. Every rule attached explicitly to the method (`blackout_rule_ids`).
2. Every global rule (`is_global = 1`).

If any rule's `dates_config` covers the current date the method is hidden.

The cache is cleared automatically on rule save / delete via the provider's `clearCache()` call. Manually flush with `bin/magento cache:clean` if you edit the underlying table by hand.

10 Price calculation

Price formation runs after the schedule + blackout filters pass.

1. Start from the shipping method's base `price`.
2. Load every Price Rule listed in `price_rule_ids` with `status = 1`.
3. For each cart item, iterate every loaded rule whose schedule allows it; evaluate the rule's conditions against the item; if it matches, compute the candidate price from the pattern.
4. Take the minimum of every candidate the rule loop produces. If at least one rule applied to a cart item the loop short-circuits — the lowest price wins.
5. Final price = `min(candidates)` or the base price when no rule applied.

10.1 Price pattern syntax

The `price` column on `extmag_exrates_price_rules` stores a short expression. Supported tokens (whitespace ignored):

| Pattern | Result |
|--------------------|--|
| empty | Skip — base price unchanged. |
| <code>20</code> | Replace base price with <code>20</code> . |
| <code>-20</code> | <code>price - 20</code> . Floored to <code>0</code> if the result would be negative. |
| <code>+20</code> | <code>price + 20</code> . |
| <code>-10%</code> | <code>price - (10 / 100) * price</code> . |
| <code>+20%</code> | <code>price + (20 / 100) * price</code> . |
| <code>10%</code> | <code>(10 / 100) * price</code> (sets the price to that percentage of the base price). |
| <code>max20</code> | <code>min(price, 20)</code> — final price capped at <code>20</code> . |
| <code>min20</code> | <code>max(price, 20)</code> — final price raised to <code>20</code> when lower. |

Patterns are case-sensitive. The numeric portion accepts an integer or decimal value in the store's base currency.

11 Customer group resolution

- **Storefront** — `Magento\Customer\Model\Session` provides the customer's group id. Guest carts default to `NOT LOGGED IN` (id `0`).

- **Admin order create** — `Magento\Backend\Model\Session\Quote` provides the quote's customer group id. The admin area is therefore treated like a logged-in customer for the purposes of method filtering.
- The carrier also enforces `area = all` for logged-in storefront customers — methods marked *Admin Only* never appear when an actual customer is checking out.

12 CSV import / export

- **Export** — `Extmag\ExtendedRates\Block\Adminhtml\System\Config\ExportShippingMethods` renders the *Export in CSV* button on the carrier configuration page. Click downloads a CSV containing every active and inactive method with all configurable fields.
- **Import** — `Extmag\ExtendedRates\Helper\ImportShippingMethods` is wired as the *Import* field's `backend_model` and processes the upload synchronously at config save time. Uploaded files land in `pub/media/extmag_sm_import/`.
- The import is atomic — either every row passes validation or the whole upload is rejected with a per-row error message.

13 Adminhtml UI components

The admin grids and edit forms are built on Magento's Admin UI Components. Three top-level listings are registered:

| Component | Purpose |
|---|--|
| <code>extmag_extended_rates_listing</code> | Shipping Methods grid (inline edit, mass status, mass delete). |
| <code>extmag_exrates_price_rules_listing</code> | Price Rules grid. |
| <code>extmag_blackout_rules_listing</code> | Blackout Rules grid. |

Per-entity edit forms (`extmag_extended_rates_form`, etc.) use Magento Rule's condition tree component and the date / time picker out of the box.

14 Hyva Theme integration

The carrier itself emits standard Magento shipping rate objects, so both the default Luma checkout and the Hyva Checkout consume them without theme-specific code. The integration shipped in commit `c996060` (release notes: *Added Hyva Theme integration*) adjusts the per-method logo rendering so it survives Hyva's checkout layout — no manual configuration is required on Hyva stores.

15 Known limitations

- **Carrier-level free shipping** is gated only by the per-method *Free Shipping* flag plus the request-level free shipping signal. The module does not introduce its own free-shipping promotion engine.
- **weekly frequency** is referenced in source code but commented out of the *Application Frequency* dropdown. Use `daily` with selected weekdays for week-based scheduling.
- **GraphQL** — no GraphQL types are registered. Shipping rates are surfaced through the standard Magento checkout GraphQL flow when the carrier is enabled, but the module's CRUD APIs are PHP-only.
- **Per-method storefront preview** — there is no built-in *preview as customer* view. Use **Debug** mode under the carrier configuration plus the `var/log/` trace to inspect why a method appears or is filtered out.