magmodules

Best Practices

Omnibus Directive for Magento 2

These are the setups we recommend for the Omnibus Directive extension, based on how shops actually announce price reductions. You'll find general do's and don'ts, a handful of real-world scenarios (a simple single-store shop, a multi-country setup, a shop that imports prices from an ERP and one that runs a lot of catalog price rules) and the mistakes we see most often. The goal is a storefront that is compliant, stays fast and shows a lowest price your customers and regulators can trust.

Recommended configurations and patterns for Omnibus Directive.

General Guidelines

Do's

- Enable the module and run `bin/magento omnibus:snapshot` well before your next sale, so there is real history to compare against
- Keep Magento cron running: it takes the daily snapshot, cleans up old history and runs the queue consumer
- Keep all price types selected so the tracked price matches what a guest pays
- Translate the label per store view
- Use "Only on Sale" as display mode unless you have a reason not to
- Check the price history grid when a customer questions a lowest price

Don'ts

- Don't truncate the price history tables to "start fresh" before a sale, you'd lose the prices the lowest price is based on
- Don't raise a price shortly before a sale to make the reduction look bigger: the module shows the earlier, lower price
- Don't disable the module per store view during a sale and re-enable it afterwards, the gap won't be recorded
- Don't exclude products just because their lowest price looks inconvenient, exclusions are meant for products that never have announced reductions

Common Scenarios

Scenario 1: Single Store, Special Prices

Use case: One store view, reductions are set with special prices on products.

Configuration:

General:

- Enabled: Yes

Pricing:

- Lookback Period: 30
- Price Types: all selected
- Track Configurable Products: Yes

Frontend:

- Display Mode: Only on Sale
- Show on Listing: Yes
- Show on Product Page: Yes

Result: Every product with an active special price shows the lowest price of the 30 days before the special price started, on product pages, category pages and in search results.

Scenario 2: Multiple Countries and Languages

Use case: One website with Dutch, Belgian (French) and German store views.

Configuration:

General (default scope):

- Enabled: Yes

Frontend (per store view):

- Label Text NL: Laagste prijs in de afgelopen 30 dagen:
- Label Text FR: Prix le plus bas des 30 derniers jours :
- Label Text DE: Niedrigster Preis der letzten 30 Tage:

Result: History is kept per store view, so each store view shows its own lowest price in its own language and currency. Prices are converted to the store view currency on display.

Scenario 3: Prices Imported from an ERP or PIM

Use case: Prices are updated every night with a CSV import or an API integration.

Configuration:

- Enabled: Yes
- Make sure the Magento cron is running, including the queue consumers (`consumers_runner`)

Result: Imports and mass updates are queued on the `omnibus.price.snapshot` queue and processed in the background, so imports stay fast. Price changes written directly into the database are still picked up by the daily snapshot at 01:30, or by the indexer when it runs "Update by Schedule".

Scenario 4: Catalog Price Rules

Use case: Reductions are made with catalog price rules, for example "20% off all jackets this weekend".

Configuration:

- Price Types: Catalog Price Rule selected
- Display Mode: Only on Sale

Result: The rule price for guests is tracked as soon as Magento has applied the rule. Note that "Save and Apply" in the admin only schedules the rules; the prices are applied, and recorded, when the indexer cron rebuilds the catalog rule index (usually within a minute). Rules with a start date are picked up by Magento's daily rule cron and the module's daily snapshot.

Scenario 5: Products That Never Get a Reduction

Use case: Gift cards, services or deposit products that should never show a lowest price.

Configuration:

- Exclude Categories: the "Gift cards" category
- Exclude SKUs: individual service SKUs
- Or: set "Hide Omnibus Lowest Price" to Yes on the product

Result: These products never show the lowest price, but their history is still recorded in case you need it later.

Performance Optimization

- The storefront reads from the `mm_omnibus_lowest_price` index only; there are no calculations during page views
- Category pages and search results load the lowest prices of all listed products in a single query
- Imports, mass updates and catalog rules are processed through the async queue
- Only price changes are stored, not a row per product per day, so the history table stays small
- For large catalogs, set the "Omnibus Lowest Price" indexer to "Update by Schedule"

Common Mistakes

Mistake: Expecting history from before the module was installed

Why it's wrong: The module can only know prices from the moment it was enabled. A product that was already reduced shows its current price as the lowest price. **Correct approach:** Enable the module and run `bin/magento`

`omnibus:snapshot` at least 30 days before an important sale.

Mistake: Testing with "Always" and forgetting to switch back

Why it's wrong: The lowest price then also shows on products that are not reduced, which confuses customers.
Correct approach: Switch back to "Only on Sale" after testing.

Mistake: The queue consumer is not running

Why it's wrong: Imports and catalog rules are queued. Without a consumer, these price changes are only recorded at the next daily snapshot. **Correct approach:** Make sure `consumers_runner` cron is enabled, or run `bin/magento queue:consumers:start omnibusPriceSnapshot` in your process manager.

Module-Specific Tips

- **Long-running sales:** History older than the lookback period plus 7 days is cleaned up. When a reduction runs longer than that, the earlier regular price eventually drops out of the window and the sale price itself becomes the lowest price. In practice a reduction that has lasted that long is treated as the new regular price.
- **Price increases:** The window always looks back from the moment the current price started. After a price increase, the lowest price is the price before the increase; with "Only when Different" this is shown, with "Only on Sale" it isn't.
- **Customer groups:** Only the prices for guest visitors (NOT LOGGED IN) are tracked. Group-specific prices are planned for a later version.
- **Bundle and grouped products** are not tracked yet.

Need More Help?

Documentation:

- All Help Articles - Complete documentation overview

Support:

- Contact Support - Get help from our team

For a complete overview of features and configuration options, see the Omnibus Directive extension on magmodules.eu

All articles for Omnibus Directive

Installation

1	Installation using Composer (recommended)
2	Installation using the Adobe Marketplace
3	Install through FTP and SSH

Configuration

1	Configuration Guide
2	Quick Start Guide

Troubleshooting

1	Troubleshooting

Background

1	**Best Practices (current)**

