GURUCODERS · PRESTASHOP MODULE

Awesome Import

User manual

Overview

awesomeimport is a commercial PrestaShop module for importing products from external supplier feeds. It supports XML, CSV, and XLS formats sourced from remote URLs or local file uploads. Features include flexible column mapping, smart transformation rules, delta (incremental) imports, dry-run previews, import history tracking, one-click rollback, and CSV export.

Requirements

- PrestaShop 1.6 or higher (1.7+ recommended for all features)
- PHP 5.6+ (PHP 7.4+ recommended)
- A valid Awesome Import license key (entered in the module configuration page)

Installation

1. Upload the `awesomeimport` folder to `<prestashop_root>/modules/` via FTP, or upload the ZIP through the PrestaShop back-office module manager.
2. Navigate to **Modules > Module Manager**, find "Awesome Import", and click **Install**.
3. After installation, go to **Modules > Awesome Import > Configuration** and enter your license key.
4. After the license is validated, the **Configure Feeds** tab becomes available.

Feed Management

Creating a Feed

1. Go to **Awesome Import > Configure Feeds**.
2. Click **Add new feed**.
3. Fill in:
    - **Name** — a human-readable label for the feed (e.g., "Supplier A Products").
    - **Source type** — URL or File (see below).
    - **Format** — CSV, XML, or XLS.

- **Primary field** — the column used to identify existing products (typically `reference` or `ean13`).
- **Default category** — products without a category mapping will be placed here.
- **Language** — the language used for multi-language text fields.

4. Save the feed.

Feed Source Types

Type	Description
URL	The module downloads the feed from a remote HTTP/HTTPS URL before each import. Supports HTTP Basic Auth (enter as `user:password` in the credentials field).
File upload	Upload a local CSV, XLS, or XML file via the feed form. The file is stored in `modules/awesomeimport/feeds/`.

After the first download or upload, the module reads a sample of rows to display the column mapping table.

Field Mapping

Once the feed has been downloaded at least once, the **Columns mapping** table appears below the feed configuration form. For each column in the feed:

1. Select the corresponding PrestaShop field from the dropdown (e.g., map feed column `PRICE` to `Price without tax`).
2. Optionally tick **Skip existing products** on a column to preserve that field for products that already exist in the catalog.
3. Save the feed to persist the mapping.

The mapping table may span multiple pages (navigate with the left/right arrows below the table) when the feed has many columns. Lines at the top of the sample that should not be imported (extra headers, totals rows, etc.) can be excluded by setting **Lines to skip**.

Smart Filter Rules

Smart rules allow you to transform or filter feed data before import without modifying the feed itself. Each rule follows the pattern:

```
IF column {COL1} {RELATION} {VALUE1} THEN column {COL2} WILL BE {VALUE2}
```

Common uses:

Goal	Rule example
Set a default supplier for all products	IF `{NO COLUMN}` EQUALS `{}` THEN `supplier` WILL BE `My Supplier`

Goal	Rule example
Deactivate out-of-stock products	IF `quantity` EQUALS `0` THEN `active` WILL BE `0`
Convert YES/NO stock to a quantity	IF `in_stock` EQUALS `YES` THEN `quantity` WILL BE `50`
Apply a price multiplier for expensive items	IF `price_tex` BIGGER THAN `1000` THEN `price_multiplier` WILL BE `1.4`
Remap category names	IF `category` EQUALS `HDD` THEN `category` WILL BE `Hard Drives`
Copy reference to supplier reference	IF `reference` DIFFERENT FROM `{}` THEN `supplier_reference` WILL BE `%s`

Rules are applied in order and can be reordered by dragging rows. Use `%s` in VALUE2 to insert the original value from the feed (optionally with a prefix/suffix: `A-%s`, `%s-SUFFIX`).

Import Modes

Full Import

A full import processes every product in the feed on each run. Existing products are updated; new products are created. This is the default mode.

- All products in the feed are evaluated regardless of whether they changed since the last import.
- Suitable for small feeds or infrequent schedules.

Delta Import (incremental)

Enable **Delta import mode** on the feed configuration form to activate incremental imports.

What "delta" means: Before processing each product, the module compares key fields between the feed row and the current database values. If nothing has changed, the product is skipped.

Fields compared during delta check:

Feed field	DB comparison
`price` / `wholesale_price`	`ps_product.price` / `wholesale_price` (rounded to 4 decimal places)
`quantity`	`StockAvailable` qty for the default combination
`description`	`ps_product_lang.description` (strip_tags, trimmed)
`active`	`ps_product.active`
`reference`	`ps_product.reference` (trimmed)

Only fields that are actually mapped in the feed are compared. If a field is not mapped, it is not part of the delta check.

When to use delta vs full:

Scenario	Recommended mode
Large catalog (1000+ products), frequent schedule (every hour)	Delta
Small catalog or infrequent schedule (daily)	Full
Feed contains price-only updates	Delta
Feed structure changes regularly	Full (to re-map)

Note: Delta mode only skips the legacy import stage for unchanged products. The feed download and parsing always run regardless of delta mode.

Dry-Run Preview

The **Test Run** button (visible when a saved feed is open) performs a simulated import of the first 10 products without writing anything to the database.

How to trigger:

1. Open a configured feed in the back-office.
2. Click the **Test Run (first 10 products, no DB save)** button in the "Feed actions" section.
3. A preview table appears below the button within seconds.

Interpreting the preview table:

Column	Meaning
Action	`create` — product does not exist yet; `update` — product already exists; `skip` — product would be skipped (disabled feed, empty primary field, or `only_new_products` mode)
Name	Product name from the feed
Reference	Value of the primary field (reference, EAN, etc.)
Price	Price as read from the feed (before multipliers)
Quantity	Stock quantity from the feed
Note	Reason for `skip` action, if applicable

The dry-run reads and parses the complete feed (to preserve the correct `objects_in_feed` count) but only presents the first 10 data rows in the preview.

Import Progress

Progress Bar

When an import is triggered from the admin panel (via the "Run now" cron links), a progress bar appears in the feed configuration page and updates every 2 seconds via AJAX. It shows:

- A percentage bar (e.g., 42%)
- A text counter: "Processed 420 of 1000 products"

The bar disappears and shows "Done" when the import finishes.

Progress is only tracked for standalone feed imports (`id_ai_feed` parameter, no `continue_if_needed` grouping). Grouped imports and multi-step imports do not update the progress bar.

What the Progress Counter Shows

The counter reflects products that have been passed to the import engine (including those skipped by smart rules or delta mode). It updates every 100 products to reduce database write pressure.

Import History

Reading the History Log

The **Import History (last 20 runs)** table appears at the bottom of each feed's configuration page. It shows the 20 most recent import runs in reverse chronological order.

Column	Description
#	Auto-increment run ID
Status	`Success` , `Error` , or `Running` (still in progress)
Started	Timestamp when the import began
Finished	Timestamp when the import completed (`-` if still running)
Created	Number of new products added
Updated	Number of existing products updated
Skipped	Number of rows skipped (invalid primary field, delta unchanged, smart rule skip, etc.)
Error	First 80 characters of any error message

Status Codes

Status	Meaning
`success`	Import completed without fatal errors
`error`	Import finished with a fatal error; check logs
`running`	Import is currently in progress (or was interrupted before completion)

A `running` status that does not change for an extended period indicates a crashed or timed-out import. In that case you can safely start a new import — the status will be overwritten.

Import Rollback

Before updating any existing product, the module saves the pre-import values of four fields to the `ai_import_rollback` table. If the import produces unwanted changes, the rollback button restores these values.

What Gets Saved for Rollback

Field	Source
`price`	`ps_product.price`
`wholesale_price`	`ps_product.wholesale_price`
`active`	`ps_product.active`
`quantity`	`StockAvailable` for the default combination (id_product_attribute = 0)

Only products that are **updated** (existing products) have rollback data saved. New products created during the import do not have a pre-import state to restore.

How to Roll Back

1. Open the feed in the back-office.
2. Click the **Rollback Last Import** button in the "Feed actions" section.
3. Confirm the dialog prompt.
4. The module restores `price`, `wholesale_price`, `active`, and `quantity` for every product that was updated during the last import, then clears the rollback buffer.

A success message confirms the rollback was applied. If no rollback data exists (e.g., it was already applied, or the last import only created new products), a warning message is shown.

Limitations

- **One rollback per feed, not per run.** The rollback buffer is per-feed and is overwritten on each new import run. You cannot roll back to an import two runs ago.

- **New products are not removed.** Rollback restores field values on updated products; it does not delete products that were created by the import.
- **Combination stock is not rolled back.** Only the stock for `id_product_attribute = 0` (the base product) is saved and restored. Combination-level stock is not captured.
- **Other fields are not rolled back.** Descriptions, names, images, categories, and all other fields changed during the import are not covered by rollback.

Cron / Scheduled Import

Cron URL (with token)

Each feed exposes two individual cron URLs in its configuration page:

URL	Behaviour
Individual feed — Start from 0	Starts a fresh import of this feed from the beginning.
Individual feed — Continue	Continues a partial import (used for time-limited hosts).

There are also four general-feed URLs that process all feeds sequentially:

URL	Behaviour
General feed — Start from 0	Imports all active feeds from the beginning
General feed — Continue	Continues multi-step import of all feeds
General feed — Start from 0 (only update)	Same as above, but skips new product creation
General feed — Continue (only update)	Continue + updates only

The cron token is a shop-specific hash. It changes if the shop domain changes. Always copy the URL from the admin panel rather than constructing it manually.

Recommended Schedule

Feed size	Recommended interval
< 500 products	Every 30 minutes
500–5 000 products (delta mode)	Every 60 minutes
5 000–50 000 products	Every 6–24 hours (full) or 60 min (delta)
> 50 000 products	Nightly full + hourly delta

Add the cron URL to your server's crontab or use an external cron service (e.g., cron-job.org):

```
*/30 * * * * curl -s "https://yourstore.com/module/awesomeimport/cronjob?
token=...&id_ai_feed=1" > /dev/null 2>&1
```

Email Notifications on Completion

Email notifications after import completion are not yet implemented in the current version. The import history table in the admin panel provides a full audit trail of each run's outcome.

Troubleshooting

Feed does not download

- Verify the URL is reachable from the server (not just from your browser).
- For HTTPS feeds, check that the server has valid SSL certificates and that PHP's `allow_url_fopen` or cURL is enabled.
- For feeds requiring HTTP authentication, enter credentials as `username:password` in the "Username / Password" field.

Products are not being created

- Ensure the feed's **Primary field** mapping matches a column that contains unique identifiers.
- Check that **Only new products** is not accidentally enabled (which disables updates but does not prevent creation — verify with a dry-run).
- Check the **Import History** for skipped product counts and any error messages.

Price is imported as 0 or incorrect

- Verify the `Price without tax` column is mapped correctly.
- If the supplier uses a comma decimal separator (e.g., `1.234,56`), set **Format prices** to the appropriate locale in the feed settings.
- Check for smart rules that may be overriding the price or price multiplier.

Delta mode is not skipping unchanged products

- Confirm that **Delta import mode** is enabled on the feed.
- Delta comparison only fires for products that already exist in the shop. New products are always processed.
- Only fields that are mapped in the current feed configuration are compared. If `price` is not mapped, price changes will not be detected.

Rollback says "No rollback data found"

- Rollback data is only saved for existing products that are updated. If the last import only created new products, there is nothing to roll back.

- The rollback buffer is cleared after each successful rollback. Running rollback a second time will show this message.

Import appears stuck (status stays "Running")

- The import may have timed out. The `running` status is updated to `success` only when the import completes normally.
- Start a new import — the old `running` entry stays in history but does not block new runs.
- On shared hosting, use the **Continue** cron URL so each request processes a batch within the PHP time limit, then calls itself again.

Dry-run shows wrong product names or prices

- The dry-run reads raw feed column values before smart rules are applied to names/prices. Smart rule output is not reflected in the dry-run preview.
- Price shown is the raw feed value; price multipliers and tax rules are not applied in the preview.

Changelog

See CHANGES.md for the full version history.

Version 1.7.0 (current)

- Delta import mode: skip unchanged products based on price, stock, description, active, and reference comparison
- Dry-run / Test mode: preview first 10 products without DB writes
- Import history: per-feed log of last 20 runs with created/updated/skipped counts
- Import rollback: restore pre-import price, wholesale price, active status, and stock quantity
- CSV export: download all products in the feed staging table as semicolon-delimited CSV
- Import progress bar: AJAX polling shows live progress during large imports
- XXE protection verified: `LIBXML_NONET` flag applied on all PHP versions