



Mass Product Updater User Guide

PrestaShop module · Compatible with PrestaShop 1.7, 8 and 9

Bulk-update prices and stock from a CSV or Excel file

Mass Product Updater imports a CSV or Excel (`.xlsx`) file and updates the price, stock, status and identifiers of your **existing** products. You save reusable column-mapping profiles, preview every change with a row-by-row **dry-run diff** before anything is written, run the import in batches with a per-row error report, and **roll back** the last run if needed. A token-protected cron lets you automate supplier feeds. No coding, no theme edits.

1. Installation

1. In the back office go to **Modules** → **Module Manager** → **Upload a module** and select the module ZIP.
2. Once installed, click **Configure** to open the setup wizard.

Updates are automatic. They are tied to your domain — no licence key to enter. Development domains (`.local` , `.test` , `localhost`) are always allowed. When a new version is available it is shown right inside the module.

2. Configuration — step by step

The Configure screen has four tabs — **Import / update**, **Mapping profiles**, **History & rollback** and **Settings**. A normal import runs as a short wizard: upload → map columns → dry-run → apply.

1. Upload a CSV or Excel file [start here](#)

On the **Import / update** tab, choose a **File (.csv / .xlsx)** and click **Upload**. Allowed types are `.csv` , `.txt` and `.xlsx` , up to 20 MB. CSV is fully supported; XLSX is read best-effort when the ZIP and SimpleXML PHP extensions are available. If you already have a saved profile, pick it from **Use a saved profile** to pre-fill the mapping and source options.

Remember: this tool **updates existing products only** — **it never creates new ones**.

This tool UPDATES existing products (price, stock, status, identifiers...). It never creates new products. CSV is fully supported; XLSX is read best-effort when the ZIP and SimpleXML PHP extensions are available.

File (.csv / .xlsx) Nessun file selezionato

1. Upload a CSV or Excel file

2. Map columns and source options

After upload, the module previews the header and the first sample rows so you can map each source column. Set the parsing options at the top, then the column mapping below.

- **Match products by** — the identifier used to find the product: *Product ID, Reference, EAN13, UPC* or *Supplier reference*.
- **Delimiter / Enclosure** — the field separator (use `\t` for a tab) and the quote character.
- **Encoding / Decimal** — the file encoding (e.g. `UTF-8`, `ISO-8859-1`, `Windows-1252`; it is converted to UTF-8) and the decimal separator (`.` or `,`).
- **First row is a header** — tick if row 1 contains column names.
- **When no product matches** — *Skip silently* or *Report in the run log*.
- **Column mapping table** — for each column, pick what it maps to: *** MATCH KEY (identifier) ***, *Price (tax excl.)*, *Wholesale price*, *Stock quantity (set)*, *Stock quantity (delta +/-)*, *Specific price (fixed)*, *Active (0/1)*, *Weight*, *Reference*, *EAN13*, *UPC*, *Minimum quantity*, or leave as *-- ignore --*.

You must map exactly one column as the **MATCH KEY** and at least one column to a field to update.

3. Dry-run preview & diff recommended

Click **Dry-run (preview)** to simulate the import without writing anything. The module reports how many rows *matched*, were *not found* or had *errors*, and opens the run detail showing a **row-by-row diff**: each changed field is displayed as ~~old~~ → new. Always review the diff before applying.

4. Run the import (apply)

When the diff looks right, click **Apply changes** and confirm. The import runs in batches (the **Batch size** set in Settings controls how many rows are processed per transaction). At the end you get a summary — *updated*, *not found*, *errors* out of the total rows — and the full run detail with the per-row log.

5. Mapping profiles reusable

On the **Mapping profiles** tab you save a column mapping once and reuse it for recurring feeds. A profile stores the **profile name**, the **match field**, the delimiter / enclosure / encoding / decimal separator, the header flag, the not-found behaviour and the per-column mapping (by 0-based column index). Saved profiles can be edited or deleted, and are selectable on the Import tab and for the cron.

6. History & rollback

The **History & rollback** tab lists recent runs with their mode (dry-run / apply), source, totals and status. Open any run to see its statistics, the per-row log (filterable by result), and a **Download report (CSV)** button. For an applied run you can click **Roll back this run** to restore the previous values of every changed field — provided the pre-update snapshot was kept (see Settings).

7. Token-protected cron automation

On the **Settings** tab, the **Scheduled import (cron)** section gives you a ready **Cron URL** containing a secret token. Call it from your scheduler (e.g. crontab) to apply the selected **Cron profile** to the configured **Cron source** — an `http(s)` URL of a supplier feed, or a plain file name placed in the module work directory (`modules/c11_csvimport/work/`). Use **Regenerate cron token** to rotate the token. The token is required and hash-checked.

8. Settings — defaults & data

The **Settings** tab also holds: **Enable the module**, **Batch size** (20–5000 rows per transaction; lower is safer on small servers), the **default source options** (delimiter, enclosure, encoding, decimal separator, header, not-found behaviour) used to pre-fill new imports, **Keep pre-update snapshot** (enables rollback — keep it on), and **Drop data on uninstall** (off by default; when on, uninstalling deletes all profiles and run history).

3. Check that it works

- Upload a small file and run a **Dry-run**; open the run detail and confirm the row-by-row diff shows the expected *old* → *new* values.
- Run **Apply changes** on a couple of rows, then open the affected products and verify the price / stock changed as shown.
- From **History & rollback**, open that applied run and click **Roll back this run**; check the products are back to their previous values.
- Paste the **Cron URL** into a browser (or call it with `curl`) — a run should appear in History against the configured profile and source.

4. FAQ

Does it create new products?

No. Mass Product Updater only updates products that already exist. Rows whose identifier does not match any product are skipped or reported, never created.

Which file formats are supported?

CSV (and tab/text `.txt`) is fully supported. Excel `.xlsx` is read on a best-effort basis when the ZIP and SimpleXML PHP extensions are available. Files up to 20 MB are accepted.

How does product matching work?

You choose a match key — Product ID, Reference, EAN13, UPC or Supplier reference — and map one column to it. Each row is matched to a product by that identifier before any field is updated.

Is the rollback safe?

Yes, as long as *Keep pre-update snapshot* is enabled (the default). The module stores the previous value of every changed field, so an applied run can restore them exactly. A run can only be rolled back once.