CSV import and export

Customer Specific Prices Documentation

Guide for Magento administrators, technical project owners, and implementation teams.

LAST UPDATED
2026-10-13

SOURCE
Open this article online

EXTENSION
Magento 2 Extension Customer Specific Prices

CSV import and export

The Import/Export package adds customer prices to Magento's standard data transfer tools. It is intended for contract price lists that are easier to maintain in a spreadsheet or another business system.

Install the package

```
composer require mageb2b/pricesystem-customerprice-
importexport:*
php bin/magento setup:upgrade
php bin/magento cache:flush
```

Prepare a file

Open **System > Data Transfer > Export** and export the customer price entity first. Use that file as the template for the target store. This avoids guessing identifiers and column names.

The customer price import supports these business fields:

Column	Meaning
`customer_id`	Numeric customer ID or account email — both always work on import. With per-website account sharing, the email is resolved within the row's website scope
`product_id`	SKU by default; numeric product ID when **Customerprice Import > Use Product ID as** is set to Product ID
`qty`	Minimum quantity
`price`	Price value
`from_date`, `to_date`	Optional active period
`price_application_type`	Fixed, nominal/percentage surcharge, or nominal/percentage discount
`price_attribute_id`	Base price source for relative calculations
`website_id`	Numeric website ID

`product_id` is a strict switch: the configured alias decides whether the column must contain SKUs or numeric IDs. For exports, **Customerprice Export** controls whether `product_id` and `customer_id` are written as SKU/email (default) or numeric IDs. Configure these aliases under **Stores > Configuration > MageB2B > Pricesystem**

before producing the template.

Import behavior

The admin import and the CLI `--behavior` option accept the standard Magento import behaviors:

Behavior	Effect
`append` / `add_update`	Validate and save the file's rows; existing records stay untouched
`replace`	**Deletes every existing customer price first**, then imports the file
`delete`	Deletes only the records that match rows in the file

`replace` wipes the whole customerprice table before importing — it is not a per-row upsert. Never run a replace import without a current export or backup.

Import safely

1. Export a small known set.
2. Keep the header row unchanged.
3. Add a few records and run the import with validation enabled.
4. Review rejected rows before importing the full file.
5. Test one fixed price, one quantity tier and one dated price in the storefront.

Date-range validation can check rows within the file and against existing customer prices. Keep it enabled unless the import is part of a controlled replacement process.

Command-line import and export

The same package also provides CLI commands for scripted or scheduled transfers:

```
php bin/magento pricesystem:import-customerprice <import_path>
php bin/magento pricesystem:export-customerprice
```

The import accepts the same columns as the admin import. The customer-price grid also offers a standard Magento export (CSV/XML) through its export button, plus a Delete mass action.

Extension points for developers

- The import dispatches the event `pricesystem_customerprice_import_before_save` before each batch is

written. Observers receive the import parameters and the insert/update row arrays.
- The import model's `checkForAdditionalValues($entityRow, $rowData)` method is intentionally public so a custom module can map extra CSV columns into the entity row through an after-plugin.

For the customer-facing CSV, see My Prices. That download is separate from Magento's administrative import and export.