Product-Customer Matrix Prices Documentation

Complete PDF Manual

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

LAST UPDATED
2026-10-13

SOURCE
Open complete documentation online

EXTENSION
Magento 2 Extension Product-Customer Matrix Prices

Table of Contents

Product-Customer Matrix Prices Documentation · productcustomermatrix

Overview
Product-Customer Matrix Prices
Getting Started
Installation
Configuration
Create a matrix
Features
Matrix pricing
Product conditions
Assignment and priority
Addons
Advanced Config
CSV import and export
REST API
GraphQL
Sample data
Reference
Configuration Fields
Database Reference
Troubleshooting
Common Issues
FAQ

Product-Customer Matrix Prices

Product-Customer Matrix defines a price once, selects products through conditions, and applies it to matching or directly assigned customers. It suits pricing agreements that cover a changing product set or a customer segment without maintaining every customer-product pair.

Pricesystem Core is included and compares the resulting matrix candidate with other price types.

How a matrix works

A matrix combines:

- price fields such as quantity, price, application type, website, validity dates and priority;
- product conditions based on product attributes, SKU or category IDs;
- customer applicability from a direct assignment or the configured customer attribute.

Product conditions support equals, not equals, in list and not in list. Multiple conditions can use AND or OR.

Assignment and priority

The default **Direct assignment first** mode prefers a current direct customer assignment over a customer-attribute match. Priority resolves ties between matches of the same type.

Global priority ignores how the customer matched and uses the higher matrix priority.

The module also adds the cart price rule condition **Customer has product customer matrix** (Yes/No) under *Cart Price Rules* > *Conditions* — for example to grant a discount to every customer who currently matches at least one matrix.

Start here

1. Install the product.
2. Review customer matching and priority configuration.
3. Create and test a matrix.
4. Read product conditions and priority resolution.

Use Customer-specific prices for direct customer-product records, Price Lists for reusable named SKU lists, or Category Price for a single Magento category.

Optional packages

- Advanced Config controls shared Pricesystem selection.

- CSV Import/Export handles matrices in bulk.
- REST API manages matrix data.
- GraphQL provides read queries.
- Sample Data installs development fixtures.

See the FAQ and troubleshooting guide.

View the product page

Installation

Pricesystem Core is included and installed as a dependency.

Replace `repo.softwaresilo.io`, `softwaresilo`, `https://repo.softwaresilo.io/` and `<token>` with the repository details shown under **Account > Licenses** in your SoftwareSilo account — see Installation & Updates.

```
composer config bearer.repo.softwaresilo.io <token>
composer config repositories.softwaresilo composer
https://repo.softwaresilo.io/
composer require mageb2b/pricesystem-productcustomermatrix:*

php bin/magento module:enable
MageB2B_PricesystemProductCustomerMatrix
php bin/magento setup:upgrade
php bin/magento setup:di:compile
php bin/magento setup:static-content:deploy
php bin/magento cache:flush
```

Check the module state:

```
php bin/magento module:status
MageB2B_PricesystemProductCustomerMatrix
```

In the Admin, open **Pricesystem > Product Customer Matrix > All Matrix Prices**. The grid should load and provide an action to create a matrix.

Continue with configuration.

Configuration

Open **Stores > Configuration > MageB2B > Pricesystem** and find the Product Customer Matrix section.

Customer account download

Enable the download link if customers should export matrix prices. **Download Mode** can produce one combined file or separate files per matrix. Set the CSV delimiter and enclosure to match the customer's spreadsheet workflow.

Customer matching

Additional Customer Attribute selects the customer attribute used for attribute-based matrix matching. The default is `group_id`. The match reads the attribute value from the customer record (`getCustomAttribute` on the CustomerInterface), so only customer attributes carried on the customer object qualify.

The module ships its own customer attribute `procustmatrix_customer_group` (label *Product Customer Matrix Group*, editable on the admin customer form), which is the intended use case for this setting: store a group value on the customer and enter the same value as the matrix's customer attribute value. A product attribute `procustmatrix_product_group` is installed as well and can be used in product conditions.

`procustmatrix_customer_group` is a protected attribute: writing it through `custom_attributes` is restricted — only admin and integration API tokens may set it, while customer and guest tokens get rejected on customer API writes. Maintain it in the Admin or through an integration; see Protected customer attributes for the shared mechanism.

Only choose an attribute whose values are populated consistently. A blank or unexpected value changes which matrices match.

Priority resolution

Mode	Result
Direct assignment first	A direct customer assignment wins over an attribute match; priority breaks ties within the same assignment type
Global priority	The higher matrix priority wins, regardless of assignment type

Direct assignment first is the default and works well for customer-specific exceptions to group-based pricing.

Duplicate validation

Custom Matrix Validation adds stricter checks for duplicate matrices and overlapping dates involving customer or attribute assignments. Enable it when the data model should reject these overlaps rather than resolve them through priority.

Shared calculation settings are documented under Pricesystem configuration.

Create a matrix

Open **Pricesystem > Product Customer Matrix > All Matrix Prices** and add a matrix.

Define the price

Enter a **Name**, **Quantity**, **Price**, **Price Application Type**, **Website**, optional **Valid From**/**Valid To** dates and a **Priority**. Relative surcharges and discounts can use a selected **Price Attribute** as their base.

Select products

Add one or more product conditions. A condition contains a source, operator and value. Supported sources include product attributes, SKU and category IDs. Operators include equals, not equals, in list and not in list.

Use AND when every condition must match. Use OR when any condition is enough. Start with one condition and confirm its product set before adding more.

Select customers

Choose either or both methods:

- assign individual customers directly, optionally with assignment dates;
- enter the customer-attribute value that should match the configured attribute.

You can also assign matrices from the other direction: the customer edit form contains a **Product Customer Matrices Assigned by Customer Relation** section with an Add-Matrix dialog and Assign/Unassign actions. Managing assignments there requires the dedicated Admin permission `MageB2B_PricesystemProductCustomerMatrix::customer_save`.

Matrix dates, website, product conditions and customer assignment must all be valid for a price to apply.

Test the matrix

Test a matching product and customer, then test one product and one customer that should not match. Check the exact quantity threshold and one unit below it. If several matrices match, test the configured priority mode.

Matrix pricing

A matrix is one price rule with a product set and a customer audience. The rule becomes eligible only when its website and date range match, the product satisfies its conditions, and the customer matches through a direct assignment or configured attribute.

The price supports fixed values, surcharges and discounts. Quantity is a minimum threshold. When several thresholds from eligible matrices are available, the closest threshold that does not exceed the requested quantity applies after matrix conflicts are resolved.

Direct customer assignments can have their own dates. These assignment dates do not replace the matrix date range; both must be current.

Once Product-Customer Matrix has produced a candidate, Pricesystem Core applies the shared price selection rule.

Product conditions

Product conditions define which catalog products belong to a matrix. Conditions can use product attributes, SKU or category IDs.

Operators

Operator	Use
Equals	Match one exact value
Not equals	Exclude one exact value
In list	Match any value in a supplied list
Not in list	Exclude values in a supplied list

Join conditions with AND when all of them must match. Join them with OR when one match is enough.

Example

To cover blue work bags but exclude a discontinued SKU, combine a color condition with a SKU exclusion using AND. To cover products in either of two categories, use an in-list category condition or separate category conditions joined with OR.

Keep values in the format stored by Magento. Attribute option IDs, labels and category IDs are not interchangeable. After saving, verify several matching and nonmatching products instead of checking only one expected result.

Assignment and priority

Several matrices may match the same customer, product and quantity. **Priority Resolution Mode** decides which match wins.

Direct assignment first

A current direct customer assignment wins over a customer-attribute match. If two direct assignments match, or two attribute matches match, the matrix with the higher numeric priority wins.

This mode is useful when group-based pricing supplies a general rule and individual customers have explicit exceptions.

Global priority

The assignment type is ignored. The matrix with the higher numeric priority wins.

Use this mode when priority alone expresses the contract hierarchy.

Testing overlaps

Create a test case in which the same customer and product match one direct matrix and one attribute matrix. Give them different priorities, then verify the result in both modes. Repeat at every shared quantity threshold that matters to the agreement.

This matrix priority is separate from the shared Pricesystem price selection, which compares the final matrix candidate with other price types.

Advanced Config

Advanced Config controls how Pricesystem compares the matrix result with other price types. Matrix assignment and matrix-to-matrix priority remain part of Product-Customer Matrix.

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

See Pricesystem Advanced Config for customer overrides, customer-group overrides and formulas. The matrix candidate uses `product_customer_matrix` in shared selection rules.

CSV import and export

The Import/Export package adds the `pricesystem_product_customer_matrix` entity to Magento's data transfer tools. It handles matrix headers, product-condition rows and direct customer assignments.

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

Export a small matrix from **System > Data Transfer > Export** and use it as the template for the target store. Keep the matrix identifier consistent across its conditions and customer assignments.

File format

The CSV uses a **hierarchical row format**: one header row per matrix, followed by optional child rows for its product conditions and customer assignments.

Matrix header columns:

Column	Meaning
`id`	Matrix identifier — set it to keep the matrix's child rows together
`name`	Matrix name
`customer_attribute_value`	Value matched against the configured customer attribute
`website_id`	Website scope
`qty`, `price`	Minimum quantity and price value
`price_application_type`	Fixed, surcharge or discount type
`price_attribute_id`	Base price source for relative calculations
`attributes_relation`	`AND` or `OR` for the product conditions
`priority`	Priority for competing matrices
`valid_from`, `valid_to`	Optional active period

Child-row columns (on their own rows beneath the matrix header):

Column	Meaning
`matrix_attribute_attribute`, `matrix_attribute_value`, `matrix_attribute_operator`	One product condition per row
`matrix_customer_customer_id`	A directly assigned customer
`matrix_customer_date_from`, `matrix_customer_date_to`	Optional assignment period

Command-line import and export

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

Verify before importing

Before a full import, verify:

- product attribute codes and values;
- SKU and category identifiers used by conditions;
- AND or OR relation;
- customer identifiers and assignment dates;
- website, quantity, application type and matrix dates.

Import one matrix first. Test a matching and nonmatching product and customer before processing the complete file.

REST API

The API package manages matrix headers, product-condition rows and direct customer assignments.

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

Resource	Base endpoint
Matrix header	`/V1/pricesystem/productcustomermatrix`
Product condition	`/V1/pricesystem/productcustomermatrixattribute`
Customer assignment	`/V1/pricesystem/productcustomermatrixcustomer`

Each resource supports:

- `GET /:id` for one record;
- `GET /search` with Magento search criteria;
- `POST` to create;
- `PUT /:id` to update;
- `DELETE /:id` to remove.

Create the matrix header before its condition and customer rows. During synchronization, remove or update child rows deliberately so stale conditions do not continue matching products.

Use a Magento integration with the Product-Customer Matrix ACL resources. Calculated price operations are documented under Pricesystem REST.

GraphQL

The GraphQL package provides read queries for matrices, product conditions and customer assignments.

```
composer require mageb2b/pricesystem-productcustomermatrix-graph-ql:*
php bin/magento setup:upgrade
php bin/magento cache:flush
```

Single record	Paginated list
`productCustomerMatrix(id)`	`productCustomerMatrices(filter, pageSize, currentPage)`
`productCustomerMatrixAttribute(id)`	`productCustomerMatrixAttributes(filter, pageSize, currentPage)`
`productCustomerMatrixCustomer(id)`	`productCustomerMatrixCustomers(filter, pageSize, currentPage)`

Filters cover stored fields such as matrix ID, name, customer attribute value, website, quantity, priority, product condition and customer assignment. Paginate list queries and request only the required fields.

The `ProductCustomerMatrixAttribute` type does not expose the condition operator (`eq`, `neq`, `in`, `nin`). The operator is stored on the record and maintained through the Admin or REST, but it is not readable through GraphQL.

For calculated storefront prices, use Pricesystem GraphQL.

Sample data

The sample packages install shared demonstration customers and products, then add matrices that exercise SKU and category conditions, supported operators, AND/OR relations and assignments. Use them only in development or disposable demos.

```
composer require mageb2b/pricesystem-sample-data:*
composer require mageb2b/pricesystem-productcustomermatrix-
sample-data:*
php bin/magento setup:upgrade
php bin/magento cache:flush
```

```
php bin/magento sampledata:verify:pricesystem-
productcustomermatrix
php bin/magento sampledata:reset:pricesystem-
productcustomermatrix
```

The reset command removes data owned by this matrix sample package. Do not run it on production. Shared fixture handling is documented under Pricesystem sample data.

Configuration Fields

This reference lists the settings exposed by `mageb2b/pricesystem-productcustomermatrix` in Magento Admin.

Configuration paths are included for controlled deployments and `config:set`. Use Magento Admin for normal changes so field validation and scope inheritance still apply. The extension status display is not listed because it reports installation state rather than a configurable value.

Product Customer Matrix

Field	Configuration path	Input	Scope
Enable download link	`pricesystem/productcustomermatrix/enable_download`	Select	Default, Website
Download Mode	`pricesystem/productcustomermatrix/download_mode`	Select	Default, Website
Csv Delimeter	`pricesystem/productcustomermatrix/csv_delimeter`	Text	Default, Website
Csv Encloser	`pricesystem/productcustomermatrix/csv_encloser`	Text	Default, Website
Additional Customer Attribute	`pricesystem/productcustomermatrix/customer_attribute`	Select	Default, Website
Priority Resolution Mode	`pricesystem/productcustomermatrix/priority_mode`	Select	Default, Website
Custom Matrix Validation	`pricesystem/productcustomermatrix/custom_matrix_validation`	Select	Default, Website

Database Reference

Product Customer Matrix separates the price rule from the product conditions and direct customer assignments that make the rule eligible.

Relationship	Why it matters
`pricesystem_product_customer_matrix_attribute` belongs to one `pricesystem_product_customer_matrix`	Defines the product attributes, categories or identifiers matched by a matrix.
`pricesystem_product_customer_matrix_customer` assigns a matrix to one `customer_entity`	Gives a named customer direct access to the matrix price.
A matrix customer-attribute value is matched against the configured customer attribute	Allows one matrix to apply to customers that share a business attribute without creating individual assignments.

The matrix itself contains the commercial rule, including website, validity, quantity and priority settings. Use the matrix editor, import add-on or APIs for updates.

Common Issues

A matrix price does not apply

Check the complete match:

1. The matrix website and date range are current.
2. The requested quantity meets the threshold.
3. The product satisfies the configured conditions.
4. The customer has a current direct assignment or the configured attribute value.
5. Any direct assignment dates are current.

If the matrix candidate exists but another storefront price appears, review Pricesystem price selection.

Too many products match

Check the AND/OR relation and each operator. An OR relation needs only one true condition. Verify whether a condition expects an option ID, raw attribute value, SKU or category ID.

A group-style match beats a direct assignment

Check **Priority Resolution Mode**. **Global priority** can allow the higher-priority attribute match to win. Use **Direct assignment first** when a direct assignment should take precedence.

A direct assignment is ignored

Confirm its customer ID and assignment dates, then check the matrix date range and website. A current assignment cannot make an expired matrix valid.

A duplicate matrix is accepted

Enable **Custom Matrix Validation** if the store should reject duplicate customer or attribute combinations and overlapping dates. Existing overlaps still need review.

An import creates unexpected matches

Export a fresh template and compare attribute codes, values, operators and relations. Test one imported matrix against both matching and nonmatching products before importing more.

FAQ

Is Pricesystem Core included?

Yes. Composer installs it as a dependency of Product-Customer Matrix.

How are products selected?

Use conditions based on product attributes, SKU or category IDs. Conditions support equals, not equals, in list and not in list, joined with AND or OR.

How are customers selected?

Assign customers directly or match the configured customer attribute. The default attribute is `group_id`.

What does Direct assignment first mean?

A current direct customer assignment wins over an attribute-based match. Priority breaks ties between matches of the same assignment type.

What does Global priority mean?

The higher matrix priority wins without regard to direct or attribute-based assignment.

Can assignments and matrices have dates?

Yes. The matrix has its own date range, and direct customer assignments can have separate dates. Both must be current.

Can customers download matrix prices?

Yes. The download can be one combined CSV or a separate file per matrix.

When should I use Category Price instead?

Use Category Price when one Magento category defines the product set. Use Product-Customer Matrix when several conditions or explicit product rules are needed.

Can I manage matrices through integrations?

Yes. CSV Import/Export and REST support write workflows. GraphQL provides read queries.