Category Specific 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 Category Specific Prices

Table of Contents

Category Specific Prices Documentation · categoryprice

Overview
Category Specific Prices
Getting Started
Installation
Configuration
Create category prices
Admin Permissions (ACL)
Features
Customer category prices
Customer-group category prices
Priority and conflicts
Cart price rule condition
Addons
Advanced Config
CSV import and export
REST API
GraphQL
Sample data
Reference
Configuration Fields
Database Reference
Troubleshooting
Common Issues
FAQ

Category Specific Prices

Category Price applies a price rule to products in a category for either one customer or one customer group. It is useful when commercial terms follow product families rather than individual SKUs.

Pricesystem Core is included with the product and decides how the resulting category-price candidate competes with other Magento and Pricesystem prices.

Two assignment types

- A customer category price applies to one customer.
- A customer-group category price applies to every customer in the selected group.

Each record contains the category, quantity threshold, value, application type, website and an optional date range. Relative surcharges and discounts can use a selected base price.

If a customer and group record both match the same quantity, the Category Price selection rule decides which record applies: group first, customer first, or a stored priority comparison that currently resolves to the customer record.

Cart price rules can also check category-price assignments: the condition **Categoryprice > Customer has categoryprice** lets a discount or promotion apply only to customers who have at least one category-price record.

Start here

1. Install the product.
2. Configure conflict handling and downloads.
3. Create a customer or group category price.
4. Review customer pricing, group pricing and conflict handling.
5. Restrict admin access with ACL permissions.

Use Customer-specific prices when a contract names individual products, Price Lists for reusable SKU lists, or Product-Customer Matrix for condition-based product selections.

Optional packages

- Advanced Config controls shared price selection.
- CSV Import/Export handles customer and group category records in bulk.
- REST API provides record management endpoints.
- GraphQL provides read queries.
- Sample Data installs development fixtures.

See the FAQ and troubleshooting guide for common questions.

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-categoryprice:*

php bin/magento module:enable MageB2B_PricesystemCategoryprice
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_PricesystemCategoryprice
```

In the Admin, open **Pricesystem > Categoryprice**. You should see separate grids for customer and customer-group category prices.

Continue with configuration.

Configuration

Open **Stores > Configuration > MageB2B > Pricesystem** and find the Categoryprice settings.

Customer versus group price

Category Price Select rule applies when a customer record and a customer-group record match the same category and quantity.

Rule	Result
Select price by priority	Compares the stored priority values and prefers the group record only when its priority is strictly higher; with equal priorities the customer record applies
Group price first	Prefers the customer-group record
Customer price first	Prefers the individual customer record

Priority selection is the default. Priority values are a technical field that admin forms, CSV import and the API do not expose, so records carry equal priority and this rule effectively prefers the customer record. The setting applies globally and cannot be overridden per website.

Choose one rule that matches the store's contract policy and test a customer who has both record types.

Customer download

The download link is enabled by default. It lets a signed-in customer export applicable customer and group category prices from the account area. The default delimiter is a semicolon and the default enclosure is a double quote.

Shared calculation settings are documented under Pricesystem configuration. Clear Magento cache after saving changes.

Create category prices

Category Price has separate records for customers and customer groups.

Customer category price

Open **Pricesystem > Categoryprice > All Customer Categoryprices** and add a record. Select the customer and category, then enter the quantity, value, price application, website and optional dates.

Customer-group category price

Open **Pricesystem > Categoryprice > All Customer Group Categoryprices**. Select the customer group instead of a customer, then complete the same price fields.

The records are also available from relevant customer, customer-group and category edit screens.

Price fields

Field	Purpose
Qty	Minimum quantity for the category price
Value	Fixed price, surcharge or discount value
Price application	Fixed, Surcharge, Surcharge %, Discount or Discount %
Base Price	Source used by a relative application type
Website	Global or website-specific scope
From Date / To Date	Optional active period

Products become eligible through Magento category membership. A product assigned to several categories can therefore match several records; see conflict handling for how the matching record is chosen, and test such products before rollout.

Test the exact quantity threshold and one unit below it. Also test an account that has both a customer and group record so the configured conflict rule is proven.

Admin Permissions (ACL)

Category Price adds granular admin permissions so you can give pricing or support users access without full admin rights.

Where To Configure Roles

In Magento Admin go to:

- **System > Permissions > User Roles** (create/edit a role)
- **System > Extensions > Integrations** (if you use integration tokens)

The permissions below show up under **Pricesystem Categoryprice All** in the role resource tree.

Admin Menu (What Maps To What)

Admin Menu	Required ACL
`Pricesystem > Categoryprice`	`MageB2B_PricesystemCategoryprice::all`
`Pricesystem > Categoryprice > All Customer Categoryprices`	`MageB2B_PricesystemCategoryprice::manage`
`Pricesystem > Categoryprice > All Customer Group Categoryprices`	`MageB2B_PricesystemCategoryprice::manage_customer_group`
`Stores > Configuration > MageB2B > Pricesystem`	`MageB2B_PricesystemCategoryprice::config`

The configuration section itself is also gated by the shared `MageB2B_PricesystemCore::config` resource.

Main ACL Resources

Permission	What It Controls
`MageB2B_PricesystemCategoryprice::manage`	Customer category-price grid and record screens

Permission	What It Controls
`MageB2B_PricesystemCategoryprice::save`	Create/update customer category prices
`MageB2B_PricesystemCategoryprice::delete`	Delete customer category prices
`MageB2B_PricesystemCategoryprice::manage_customer_group`	Customer-group category-price grid and record screens
`MageB2B_PricesystemCategoryprice::save_customer_group`	Create/update customer-group category prices
`MageB2B_PricesystemCategoryprice::delete_customer_group`	Delete customer-group category prices
`MageB2B_PricesystemCategoryprice::config`	Categoryprice configuration section

Tip: Align ACL With REST API Access

If you use the **REST API Add-On** (package `mageb2b/pricesystem-categoryprice-api`), the same resources protect its routes:

- Read and search routes require `manage` or `manage_customer_group`
- Create and update routes require `save` or `save_customer_group`
- Delete routes require `delete` or `delete_customer_group`

Keep the integration user's permissions minimal. See REST API for installation and the endpoint reference.

Customer category prices

A customer category price gives one customer commercial terms for products assigned to a category. It avoids creating a separate row for every product when the agreement follows a product family.

Create records from the central customer category-price grid, a customer edit screen or a category edit screen. Each record can have quantity tiers, website scope and dates.

Example

Northstar Industrial receives a 7 percent discount on products in the Work Bags category from a quantity of ten. Create a customer category record with quantity 10, application type **Discount %**, value 7 and the intended base price.

Products must belong to the selected category in the current catalog. If a product belongs to several matching categories, the conflict rules determine which matching record applies within this price type.

If the customer's group also has a matching category price at the same quantity, the configured customer versus group rule decides which record applies.

Customer-group category prices

A customer-group category price applies the same category terms to every customer in a Magento customer group. It suits broad policies such as wholesale or dealer pricing.

Create records from the customer-group category-price grid, a customer-group edit screen or a category edit screen. The record supports quantity, price application, base price, website and dates.

Group membership comes from the customer's current Magento customer group. If the customer changes group, the applicable group category prices change with it.

Guest pricing can use Magento's NOT LOGGED IN group when a record is created for that group. Test guest catalog and cart behavior separately from signed-in customers.

When an individual customer record also matches, the Category Price Select rule decides whether the group or the customer record applies.

Priority and conflicts

Conflicts appear in two situations: a product belongs to several matching categories, or both a customer record and a customer-group record apply to the same quantity.

Same record type

When several category records of the same type match at the same quantity — for example a product assigned to two discounted categories — the stored priority decides first. Because all records carry the same default priority, the shared Pricesystem price selection then applies and prefers the lower resulting price by default.

Customer record versus group record

The Category Price Select rule decides which record type applies:

- **Group price first** always prefers the customer-group record.
- **Customer price first** always prefers the individual customer record.
- **Select price by priority** compares the stored priority values and only prefers the group record when its priority is strictly higher. Records created in the admin, by CSV import or through the API all carry the same priority, so this rule resolves to the customer record in practice.

Do not assume a more specific Magento category automatically wins. Test products that are assigned to multiple matching categories with an account that has both record types.

After Category Price selects its own candidate, the shared Pricesystem selection rule decides how it competes with other price types.

Cart price rule condition

Category Price registers the condition **Categoryprice > Customer has categoryprice** under *Marketing > Cart Price Rules > Conditions*. It is a Yes/No check on the quote's customer, so a promotion can apply only to accounts that already carry category pricing.

The condition is met when at least one currently valid record exists for the customer:

- a **customer** category price assigned to that customer, or
- a **customer-group** category price assigned to the customer's current group.

Records outside their **From**/**To** date range do not count. The check does not look at the cart contents — it only verifies that some matching record exists.

A typical use is a small extra discount for every contract customer without maintaining a separate customer segment. Test the rule with a customer whose only records are expired — they must not receive the discount.

Advanced Config

Advanced Config controls the shared Pricesystem selection rule for the system, customer groups and individual customers. It does not replace the Category Price rule that chooses between customer and group category records.

```
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. Category Price participates under the `categoryprice` code.

CSV import and export

The Import/Export package adds separate Magento data transfer entities for customer category prices and customer-group category prices.

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

Export each entity from **System > Data Transfer > Export** and use the resulting file as its template. Customer rows identify a customer; group rows identify a customer group. Both include the category, quantity, value, application type, optional base price, website and dates.

The customer entity uses the columns `customer_id`, `category_id`, `qty`, `price`, `price_application_type`, `price_attribute_id`, `website_id`, `from_date` and `to_date`. The group entity uses `group_id` instead of `customer_id` with the same remaining columns.

Identifiers

Column	Accepted values
`customer_id`	Numeric customer ID or the account email — both always work on import. With per-website account sharing, the email is resolved within the row's website scope
`group_id`	Numeric customer group ID or the customer group code (for example `Wholesale`)
`category_id`	Numeric category ID. The group entity additionally accepts a slash-separated category name path
`website_id`	Numeric website ID

There is no product or SKU column — eligibility comes from Magento category membership. For exports, **Categoryprice Export > Use Customer ID as** under **Stores > Configuration > MageB2B > Pricesystem** controls whether `customer_id` is written as the email address (default) or the numeric ID.

Import behavior

The admin import (**System > Data Transfer > 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 record of the entity first**, then imports the file
`delete`	Deletes only the records that match rows in the file

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

Command line import and export

The package also registers console commands for automated transfers:

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

Imports accept a required file path plus `--behavior`, `--field_separator`, `--field_multiple_value_separator`, `--fields_enclosure` and `--delete_file_after_import`. Exports write to `var/export/` by default and accept `--export_path`, `--field_separator` and `--fields_enclosure`.

Both admin grids additionally offer an **Export** mass action for selected rows.

Test one customer record and one group record before importing a full file. After import, verify a product in the selected category, the exact quantity threshold and a customer for whom both record types match.

REST API

The API package exposes separate CRUD and search routes for customer and customer-group category prices.

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

Resource	Method and endpoint
Read customer record	`GET /V1/pricesystem/categoryprice/:id`
Search customer records	`GET /V1/pricesystem/categoryprice/search`
Create customer record	`POST /V1/pricesystem/categoryprice`
Update customer record	`PUT /V1/pricesystem/categoryprice/:id`
Delete customer record	`DELETE /V1/pricesystem/categoryprice/:id`
Read group record	`GET /V1/pricesystem/categoryprice/customergroup/:id`
Search group records	`GET /V1/pricesystem/categoryprice/customergroup/search`
Create group record	`POST /V1/pricesystem/categoryprice/customergroup`
Update group record	`PUT /V1/pricesystem/categoryprice/customergroup/:id`
Delete group record	`DELETE /V1/pricesystem/categoryprice/customergroup/:id`

Search routes accept Magento `searchCriteria` parameters. Use a Magento integration

with the matching Category Price ACL permissions: `manage`/`manage_customer_group` for reads, `save`/`save_customer_group` for writes and `delete`/`delete_customer_group` for deletes.

These endpoints manage records. The Pricesystem REST API provides calculated-price operations.

GraphQL

The GraphQL package adds read queries for customer and customer-group category price records.

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

Single record	Paginated list
`categoryprice(id)`	`categoryprices(filter, pageSize, currentPage)`
`categorypriceCustomergroup(id)`	`categorypriceCustomergroups(filter, pageSize, currentPage)`

Filters can target fields such as category entity ID, customer or group ID, quantity, website and date. Request only the required fields and paginate lists.

All four queries require an **admin or integration token**. Customer and guest calls are rejected with an authorization error — storefronts should not call these queries.

For a calculated storefront price rather than stored records, use the shared Pricesystem GraphQL package.

Sample data

The sample packages create shared demonstration customers, products and categories, then add customer and group category price examples. They are intended for development and disposable demos.

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

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

The reset command removes records owned by the Category Price sample package. Do not run it against production data. See Pricesystem sample data for the shared fixtures.

Configuration Fields

This reference lists the Magento Admin settings exposed by the base package and its documented add-ons: `mageb2b/pricesystem-categoryprice`, `mageb2b/pricesystem-categoryprice-importexport`.

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.

`mageb2b/pricesystem-categoryprice`

Categoryprice settings

Field	Configuration path	Input	Scope
Category Price Select rule	`pricesystem/categoryprice/price_select_rule`	Select	Default
Enable download link	`pricesystem/categoryprice/enable_download`	Select	Default, Website
Csv Delimeter	`pricesystem/categoryprice/csv_delimeter`	Text	Default, Website
Csv Encloser	`pricesystem/categoryprice/csv_encloser`	Text	Default, Website

`mageb2b/pricesystem-categoryprice-importexport`

Categoryprice Export

Field	Configuration path	Input	Scope
Use Customer ID as	`pricesystem/categoryprice_export/customer_id_alias`	Select	Default, Website

Categoryprice Import

Field	Configuration path	Input	Scope
Enable Date Range Validation	`pricesystem/categoryprice_import/enable_date_range_validation`	Select	Default, Website

Database Reference

Category Price stores customer-specific and customer-group-specific pricing against Magento categories. The relationships below show how those prices are scoped and resolved.

Relationship	Why it matters
`pricesystem_categoryprice` assigns a price to one `catalog_category_entity`	Defines the category whose products can inherit the customer price.
`pricesystem_categoryprice` applies to one `customer_entity`	Limits the price to an individual customer account.
`pricesystem_categoryprice` belongs to one `store_website`	Keeps the price within the website where it is valid.
`pricesystem_categoryprice` can use one `eav_attribute`	Selects a product price attribute when the rule should not use the standard price.
`pricesystem_categoryprice_customergroup` links one `catalog_category_entity` with one `customer_group`	Provides the same category-based model for an entire customer group.
`pricesystem_categoryprice_customergroup` belongs to one `store_website`	Separates group pricing between websites.

The table intentionally omits storage columns and indexes. Manage prices through Magento Admin, imports or the module APIs rather than writing directly to these tables.

Common Issues

A product does not receive the category price

Confirm the product is assigned to the selected Magento category. Then check the customer or group, quantity threshold, website, date range and module status.

If Category Price calculates a candidate but the storefront shows another price, review Pricesystem price selection.

The wrong category record wins

Look for other categories assigned to the product. Several records of the same type can match at the same quantity; because all records carry the same stored priority, the shared Pricesystem price selection applies the lower resulting price by default.

The group price wins over a customer price

Check **Category Price Select rule**. **Group price first** deliberately prefers the group record. **Customer price first** prefers the customer record. **Select price by priority** only prefers the group record when its stored priority is strictly higher, which admin-created records never are — it resolves to the customer record in practice.

A guest price does not appear

Verify that the record belongs to Magento's NOT LOGGED IN group, and test without a signed-in customer session. Website, dates and quantity still apply.

Downloaded prices are missing

Confirm the download link is enabled and the customer has current records for the active website. Future and expired records are not included as active prices.

Imported records are rejected

Export a fresh template for the correct entity type. Check category, customer or group identifiers, website, quantity, dates and application type. Test customer and group files separately.

FAQ

Is Pricesystem Core included?

Yes. It is installed as a dependency of Category Price.

Does the price apply to every product in a category?

It applies to products whose Magento category assignments match the selected category. Products in several categories can match several records, so test which record applies before rollout.

Can I price a category for one customer?

Yes. Create a customer category price. Customer-group category prices are a separate record type.

What happens when customer and group prices both match?

Choose **Select price by priority**, **Group price first** or **Customer price first** in configuration. Priority selection compares the stored record priorities; since admin, import and API never set them, it resolves to the customer record in practice.

Are quantity thresholds supported?

Yes. Create records with different minimum quantities. The closest eligible quantity threshold applies.

Can the rule be a discount rather than a fixed amount?

Yes. Application types include fixed price, nominal and percentage surcharge, and nominal and percentage discount.

Can category prices be scheduled?

Yes. Records support optional start and end dates plus website scope.

Can customers download their category prices?

Yes. Enable the customer download link. The export includes currently applicable customer and group category prices.

When is Product-Customer Matrix a better fit?

Use Product-Customer Matrix when products should be selected through several conditions or a direct product set rather than one Magento category.