Price Lists 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 Price Lists

Table of Contents

Price Lists Documentation · pricelist

Overview
Price Lists
Getting Started
Installation
Configuration
Create a price list
Features
Named price lists
Automatic assignment
Priority and quantity merging
Addons
Advanced Config
CSV import and export
REST API
GraphQL
Sample data
Reference
Configuration Fields
Database Reference
Troubleshooting
Common Issues
FAQ

Price Lists

Price Lists group product prices into named lists that can be assigned to customers or customer groups. A list can have its own website, active period and priority, which makes it suitable for contracts, dealer levels and regional assortments.

Pricesystem Core is included with the product. It calculates the final storefront price and provides shared features such as My Prices and price selection.

How a price list is structured

A price list contains two kinds of data:

- the list itself, including its name, status, priority, website and optional dates;
- product rows, each with a product, quantity, value and application type.

Customers can receive a list through direct assignment, customer-group assignment or matching customer and address attributes. A list marked **Is Base Pricelist** becomes a separate Price System price type. With **Pricelist** placed before **Base Pricelist** in Price Priority Order, it can supply a price when no regular list has a matching row.

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

When to use Price Lists

Use Price Lists when the same set of product prices should be reused across customers. Use Customer-specific prices for a one-customer, one-product agreement, Category Price for category rules, or Product-Customer Matrix for condition-based product sets and customer matching.

Start here

1. Install the product.
2. Review the configuration.
3. Create a price list and add products.
4. Choose manual or automatic assignment.
5. Test priority resolution when more than one list can match.

The customer account can show and export applicable prices through My Prices.

Optional packages

- Advanced Config controls shared selection rules.
- CSV Import/Export handles larger data sets.

- REST API manages lists, products and assignments.
- GraphQL provides read queries.
- Sample Data creates 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. You do not need to purchase or require it separately.

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-pricelist:*

php bin/magento module:enable MageB2B_PricesystemPricelist
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_PricesystemPricelist
```

In the Magento Admin, open **Pricesystem > Pricelist > All Pricelist**. The grid should load and allow you to create a list.

Continue with configuration.

Configuration

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

Customer download

Setting	Default	Purpose
Enable download link	Yes	Lets signed-in customers download applicable list prices
CSV delimiter	Semicolon (`;`)	Separates export columns
CSV encloser	Double quote (`"`)	Quotes text fields when needed

Product prices and list names

- **Can product have multi qty price** allows several quantity thresholds for one product in a list. It is enabled by default.
- **Use random name for pricelist if not supplied** generates a name when the admin or API request leaves it empty. It is enabled by default.

Assignment behavior

- **Deactivate old pricelist** deactivates a customer's previous list when a new one is assigned. It is disabled by default.
- **Enable auto-assignment of customers to a pricelist** evaluates customer and address attributes when a customer registers in the storefront. It does not re-evaluate existing customers. It is disabled by default.
- **Strict comparison** changes attribute matching from loose to strict comparison.

Enable automatic assignment only after testing representative customer records. Differences in data type, capitalization or formatting can change a match.

Multiple matching lists

- **Merge pricelist qtys** combines quantity rows from applicable lists. If disabled, lower-priority lists are ignored. Lists tied at the highest priority can still contribute quantity rows.
- **Merge base pricelist under Pricelist in My Prices** combines base-list rows with regular list rows in the customer account.
- **Enable base pricelist validation** limits a website to one list marked **Is Base Pricelist**. Keep it enabled so the Base Pricelist price type has one unambiguous source per website.

Admin performance

Auto-load Catalog Products on pricelist edit is disabled by default. Keep it disabled on large catalogs and load the product grid only when needed.

Shared calculation settings are covered in Pricesystem configuration. Clear Magento cache after saving and test one assigned customer before importing a full price set.

Create a price list

Open **Pricesystem > Pricelist > All Pricelist** and add a new list.

Define the list

Enter a clear name, enable the list, choose its website and set its priority. Start and end dates are optional. Mark a list as **Is Base Pricelist** only when it should provide the separate Base Pricelist price type. It is not linked to an individual regular list.

A higher priority matters when several applicable lists contain a price for the same product and quantity. The merge setting determines whether lower-priority lists may still contribute other quantity thresholds.

Add products and prices

Save the list, then open its product section. Add each product with:

- product;
- minimum quantity;
- price value;
- application type;
- optional base price source for a surcharge or discount.

Application types are Fixed, Surcharge, Surcharge %, Discount and Discount %. A fixed value is the final unit price. Surcharge types add to the selected base price (flat amount or percent), Discount types subtract from it.

If multiple quantity prices are enabled, add one row for each threshold. At a requested quantity, the closest eligible threshold applies.

Assign customers

Use one or more of these methods:

1. Assign individual customers from the list or customer record.
2. Assign a customer group when everyone in that group should receive the list.
3. Configure customer or address attribute matching and enable automatic assignment.

Direct and group assignments are easier to audit. Attribute matching is useful for rules such as dealer class or country — but note that it is evaluated **only when a customer registers in the storefront**. There is no scheduled re-evaluation, and existing customers are never matched retroactively; assign those manually or through the API. The address condition is checked against the customer's default billing address only.

Test the result

Sign in as one assigned customer and one unassigned customer. Check a product at the base quantity and at every configured threshold. If another list can match, test that overlap as well.

The final price still follows the shared Pricesystem selection rule.

If you use a base list as a fallback, select **Price Priority Order** and place **Pricelist** before **Base Pricelist**. Test one product contained in the regular list and another available only from the base list.

Named price lists

A named price list keeps a reusable set of product prices together. You can activate or deactivate the whole set, schedule it, limit it to one website and assign it to several customers or groups.

List fields

Field	Purpose
Name	Identifies the agreement in the Admin and integrations
Active	Makes the list eligible for price calculation
Priority	Resolves competing applicable lists
Website	Limits the list to one website
From / To	Optional active period
Is Base Pricelist	Exposes the list as the separate Base Pricelist price type
Customer and address attributes	Define optional automatic assignment conditions

Product rows have their own quantity and price fields. A list can therefore contain several breaks for one SKU when multi-quantity prices are enabled. On the list edit page, the assigned-products grid supports inline row editing for quick quantity and price corrections.

Use names that staff can connect to a contract or customer segment. Generated names are useful for API-created lists, but a stable business reference is usually easier to support.

Assign customers and groups

Assignments can be maintained from three places:

- the price list's own assignment sections when you edit the list;
- the customer edit form, which shows **Pricelists Assigned by Customer Relation** and **Pricelists Assigned by Customer Group Relation** (the latter shows what the customer inherits through their group);
- the customer-group edit form, which lists the group's assigned lists.

The price-list grid also offers a **Copy pricelist(s)** mass action. It duplicates the selected lists — useful as a starting point for a seasonal or successor list before adjusting product rows. The grid shows the number of assigned customers and customer groups per list and can filter by assigned customer email or group. Its **Export** mass action writes the selected

lists in the normalized CSV format used by the Import/Export package.

A base list is not attached to a particular regular list. It participates in Price System as its own price type. To use it as a fallback, put **Pricelist** before **Base Pricelist** in the global or customer-specific Price Priority Order. If the regular list has no valid row for the product and quantity, evaluation continues to the base list.

Other selection strategies treat both types as normal price candidates. Under Lowest Price, for example, the base-list value may win even when a regular list also has a value. Keep base-list validation enabled so a website has only one list marked for this role.

Automatic assignment

Price Lists can match customers using a customer attribute, an address attribute, or a combination of both. This allows rules such as a dealer class plus billing country without maintaining every customer assignment by hand.

Important — when matching runs: automatic assignment is evaluated only when a customer **registers in the storefront** (`customer_register_success`). It does not run when an administrator creates or edits a customer, and there is no scheduled re-evaluation. Existing customers are never auto-assigned — assign them manually or through the API.

Also note: the **address attribute is checked against the customer's default billing address only**. A matching value on a different address does not trigger assignment.

Configure matching

1. Edit the price list.
2. Choose the customer attribute and expected value, if needed.
3. Choose the address attribute and expected value, if needed.
4. Set whether both conditions must match.
5. Enable automatic assignment under **Stores > Configuration > MageB2B > Pricesystem**.

The strict comparison setting also checks value types. Use it only when the stored customer data is consistent with the configured comparison value.

Test before enabling it broadly

Prepare at least these cases:

- a customer who should match;
- a customer who should not match;
- a customer with a missing address value;
- a customer whose value differs only by formatting or type.

Review the assigned list and storefront price for each case. Manual customer and group assignments remain available for exceptions or contracts that should not depend on profile data.

Priority and quantity merging

Priority decides which applicable price list supplies a product price when more than one list matches. Higher-priority lists are considered before lower-priority lists.

The **Merge pricelist qtys** setting changes what happens across quantity thresholds:

- When enabled, applicable lists can contribute quantity rows.
- When disabled, rows from lower-priority lists are ignored.
- Lists tied at the highest priority can still combine their quantity rows.

Within the resulting rows, the closest eligible quantity threshold applies. For a requested quantity of 24, a threshold of 10 applies before a threshold of 1, while a threshold of 50 is not eligible.

Example

An assigned contract list has priority 20 and a quantity-1 price. A dealer list has priority 10 and a quantity-50 price. With merging enabled, the dealer list may supply the quantity-50 row. With merging disabled, only the contract list contributes because it has higher priority.

Test overlaps with the exact quantities customers order. A list-priority check at quantity 1 does not prove that quantity 50 resolves the same way.

After Price Lists supplies its candidate, the shared Pricesystem selection rule decides how that candidate competes with other Magento and Pricesystem price types.

Advanced Config

Advanced Config controls how Pricesystem chooses among available price types. It can set selection rules and priority lists for individual customers or customer groups, and it can calculate a result from a formula.

These are shared Pricesystem rules. Price List still determines which lists and product rows apply.

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

Use the Advanced Config guide for setup and precedence. Price List participates in these rules under the `pricelist` price code.

CSV import and export

The Import/Export package adds two entities to Magento's data transfer tools:

Entity	Format
`pricesystem_pricelist` (Pricesystem Pricelist)	Normalized — one row per customer assignment, customer-group assignment or product row. The identifier columns for products, customers and groups are configurable.
`pricesystem_pricelist_legacy` (Pricesystem Pricelist (Legacy))	The previous packed format: customer and group assignments travel in packed `user_ids` and `group_ids` columns on the list header row.

Prefer the normalized entity for new integrations and keep the legacy entity for files produced by older package versions. Either export can contain list headers, product rows, customer assignments and customer-group assignments.

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

Export a small list from **System > Data Transfer > Export** and use it as the template. The default aliases make products readable as SKUs and customers readable as email addresses. The **Export** mass action on the price-list grid produces the same normalized format, so a grid export also works as an import template.

When importing:

1. Keep each list's identifying fields consistent across its rows.
2. Confirm product SKUs, customer emails and group IDs in the target store.
3. Test a small file with one product and assignment.
4. Review validation errors before importing the full set.
5. Check the list in the Admin and storefront after import.

The package also provides CLI import and export commands:

```
php bin/magento pricesystem:import-pricelist
/path/to/pricelist.csv
php bin/magento pricesystem:export-pricelist
```

Run `--help` on either command for the arguments supported by the installed package.

REST API

The API package manages four resources: price lists, list products, customer assignments and customer-group assignments.

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

Resource routes

Resource	Base route	Operations
Price lists	`/V1/pricesystem/pricelist`	Get, search, create, update and delete
List products	`/V1/pricesystem/pricelistproduct`	Get, search, create, update and delete
Customer assignments	`/V1/pricesystem/pricelistcustomer`	Get, search, create, update and delete
Group assignments	`/V1/pricesystem/pricelistgroup`	Get, search, create, update and delete

Single-record reads, updates and deletes append `/:id`. Search routes append `/search` and accept Magento `searchCriteria` parameters.

Bulk list creation uses `POST /V1/pricesystem/pricelistbulk`. Product rows support bulk create and bulk update through `/V1/pricesystem/pricelistproductbulk`. Customer assignments can also be removed by customer ID through `/V1/pricesystem/pricelistcustomer/deleteByCustomerId/:customerId`.

Embedded assignments

A price-list save payload can embed `products`, `customers` and `groups` directly instead of posting each relation through its own resource. The save response reports rejected identifiers in `invalid_product_ids`, `invalid_customer_ids` and `invalid_group_ids`, so read those fields even when the request itself succeeds.

Product identifiers in API payloads are resolved strictly through the configured API product attribute (`pricesystem/api/product_attribute`, active when `pricesystem/api/use_custom_api_fields` is enabled) — there is no implicit `entity_id` fallback. Submit the identifier type the store is configured for; unresolved identifiers are rejected with a validation error.

`DELETE /V1/pricesystem/pricelist/all` removes **all** price lists. It exists for

migration and reset scenarios — protect the `MageB2B_PricesystemPricelist` delete ACL accordingly and never call it as a cleanup shortcut.

Create a Magento integration with only the required Price List ACL resources. For synchronization jobs, process each bulk response and log rejected records. An HTTP success does not prove that every submitted item passed validation.

Calculated price endpoints belong to Pricesystem REST. The routes above maintain Price List data.

GraphQL

The GraphQL package provides read queries for lists, products and assignments.

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

Single record	Paginated list
`pricelist(id)`	`pricelists(filter, pageSize, currentPage)`
`pricelistProduct(id)`	`pricelistProducts(filter, pageSize, currentPage)`
`pricelistCustomer(id)`	`pricelistCustomers(filter, pageSize, currentPage)`
`pricelistGroup(id)`	`pricelistGroups(filter, pageSize, currentPage)`

Filters use standard GraphQL filter inputs for fields such as list ID, product ID, customer ID, group ID and active state. Paginate list queries and request only the fields needed by the client.

These queries expose stored Price List records. For calculated storefront prices, use the shared Pricesystem GraphQL package.

Sample data

The sample-data packages install shared demonstration customers and products, then add example lists, product rows and assignments. Use them only in development or disposable demo environments.

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

Verify the Price List fixtures:

```
php bin/magento sampledata:verify:pricesystem-pricelist
```

Reset only the Price List sample set:

```
php bin/magento sampledata:reset:pricesystem-pricelist
```

The reset command removes data owned by this sample package. Run it only after confirming that the environment is disposable. Shared fixture handling is documented under Pricesystem sample data.

Configuration Fields

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

Pricelist settings

Field	Configuration path	Input	Scope
Enable download link	`pricesystem/pricelist/enable_download`	Select	Default, Website
Csv Delimeter	`pricesystem/pricelist/csv_delimeter`	Text	Default, Website
Csv Encloser	`pricesystem/pricelist/csv_encloser`	Text	Default, Website
Can product have multi qty price	`pricesystem/pricelist/can_product_have_multi_qty_price`	Select	Default, Website
Use random name for pricelist if not supplied	`pricesystem/pricelist/use_random_pricelistname`	Select	Default, Website
Deactivate old pricelist	`pricesystem/pricelist/deactivate_old_pricelists`	Select	Default, Website
Enable auto-assigment of customers to a pricelist	`pricesystem/pricelist/enable_auto_assign_customers_to_pricelist`	Select	Default, Website
If enabled, strict comparison (===) is used for matching; otherwise, loose comparison (==) is applied.	`pricesystem/pricelist/match_exact_type`	Select	Default, Website
Merge pricelist qtys	`pricesystem/pricelist/merge_pricelist_qtys`	Select	Default, Website
Merge base pricelist under Pricelist in My Prices	`pricesystem/pricelist/merge_base_pricelist_in_my_prices`	Select	Default, Website
Enable base pricelist validation	`pricesystem/pricelist/enable_base_pricelist_validation`	Select	Default, Website

Field	Configuration path	Input	Scope
Auto-load Catalog Products on pricelist edit	`pricesystem/pricelist/auto_load_catalog_products_in_admin`	Select	Default

Visibility dependencies:

- `pricesystem/pricelist/match_exact_type` is available when `pricesystem/pricelist/enable_auto_assign_customers_to_pricelist` is `1`.

`mageb2b/pricesystem-pricelist-importexport`

Pricelist Import

Field	Configuration path	Input	Scope
Use Product ID as	`pricesystem/pricelist_import/product_id_alias`	Select	Default, Website
Enable Date Range Validation	`pricesystem/pricelist_import/enable_date_range_validation`	Select	Default, Website
Use Customer ID as	`pricesystem/pricelist_import/customer_id_alias`	Select	Default, Website
Use Group ID as	`pricesystem/pricelist_import/group_id_alias`	Select	Default, Website

Pricelist Export

Field	Configuration path	Input	Scope
Use Product ID as	`pricesystem/pricelist_export/product_id_alias`	Select	Default, Website
Use Customer ID as	`pricesystem/pricelist_export/customer_id_alias`	Select	Default, Website
Use Group ID as	`pricesystem/pricelist_export/group_id_alias`	Select	Default, Website

Database Reference

A price list is a reusable set of product prices with its own website scope and customer audience. Its assignments are stored separately so the same list can serve several customers or groups.

Relationship	Why it matters
`pricesystem_pricelist` belongs to one `store_website`	Defines where the list and its currency context are valid.
`pricesystem_pricelist_product` links one `pricesystem_pricelist` with one `catalog_product_entity`	Stores the product covered by a list entry.
`pricesystem_pricelist_product` can use one `eav_attribute`	Allows a configured product price attribute to supply the list value.
`pricesystem_pricelist_customer` assigns a list to one `customer_entity`	Supports negotiated lists for individual customer accounts.
`pricesystem_pricelist_group` assigns a list to one `customer_group`	Reuses a list for an entire Magento customer group.

Assignment and product records belong to the parent price list. Use Magento Admin, the import add-on or the module API so validation and price cache invalidation are handled correctly.

Common Issues

A customer does not receive the list

Check that the list is active, belongs to the current website and is within its date range. Then verify the direct customer assignment, group assignment or attribute-matching rule.

If automatic assignment is enabled, compare the stored customer and address values with the configured values. Strict comparison also requires matching data types.

A product has the wrong list price

Confirm that the product row exists and that the requested quantity meets its threshold. Then review all other applicable lists, their priorities and **Merge pricelist qtys**.

If the Price List candidate is correct but the storefront uses another price, inspect the shared Pricesystem selection rule.

A lower-priority list still contributes a tier

This is expected when quantity merging is enabled. Disable **Merge pricelist qtys** if only the highest-priority list should supply rows. Lists tied at the highest priority can still combine quantity rows.

A base-list price is missing

Confirm that the base list is active, belongs to the website, contains the product and applies to the customer's assignment or group. Then check the Price System strategy. For fallback behavior, **Pricelist** must come before **Base Pricelist** in the applicable Price Priority Order. Base-list validation should also remain enabled so another list cannot claim the same role on that website.

The Admin product grid is slow

Keep **Auto-load Catalog Products on pricelist edit** disabled. Open the catalog product grid only when you need to add or inspect products.

An import fails

Export a fresh template from the target store. Check list identifiers, SKUs, customer aliases, group IDs, dates and application types. Import a small file first and review Magento's validation report.

FAQ

Is Pricesystem Core included?

Yes. Composer installs the shared core with Price Lists. It does not require a separate purchase.

Can one list be assigned to several customers?

Yes. Assign customers individually, assign a customer group, or match customer and address attributes.

What happens when several lists match?

List priority and the quantity-merge setting decide which product rows are available. Higher-priority lists win when merging is disabled. With merging enabled, applicable lists can contribute quantity rows.

What is a base price list?

A base price list is a Price List marked **Is Base Pricelist**. Price System treats it as the separate price type **Base Pricelist**, not as a child of a regular list.

This distinction matters when you configure price selection. With **Price Priority Order**, place **Pricelist** before **Base Pricelist** to get the usual fallback behavior. Price System then uses a regular list when it has an applicable product price and continues to the base list when it does not. With **Lowest Price** or **Highest Price**, both types are ordinary candidates and either one may win when both provide a value.

For example, a dealer list can contain a negotiated price for Product A while the base list contains Product A and Product B. If the priority order is Pricelist, then Base Pricelist, the dealer receives the negotiated price for Product A and the base-list price for Product B.

The base list must still be active, valid for the website and date, and applicable to the customer or customer group. Keep base-list validation enabled so only one list per website can claim this role.

Can lists start and end automatically?

Yes. Set optional start and end dates on the list. Only active lists within their date window take part in calculation.

Can a product have quantity prices inside a list?

Yes, when multi-quantity prices are enabled. Add a row for each minimum quantity. The closest eligible threshold applies.

Can customers download their prices?

Yes. Enable the customer download link. The My Prices page can merge base-list prices into the Price List section or show them separately, depending on configuration.

This setting changes only how prices are grouped in **My Prices**. It does not merge the two price types during calculation.

When should I use Customer-specific prices instead?

Use Customer-specific prices when one agreement names a particular customer and product. Use Price Lists when a reusable set should be assigned to several customers or groups.

Can I synchronize lists from another system?

Yes. Use Magento CSV Import/Export or the REST API. GraphQL provides read queries.