



SOFTWARESILOS

Pricelist

PDF Manual

MODULE

pricelist

LAST UPDATED

2026-03-29

Table of Contents

Pricelist · pricelist

- [Overview](#)
- [Pricelist](#)
- Getting Started
 - [Installation & Setup](#)
 - [Creating Pricelists](#)
- Features
 - [Automatic Customer Assignment](#)
 - [Named Pricelist Management](#)
 - [Priority-Based Resolution](#)
- Addons
 - [Advanced Config](#)
 - [CSV Import/Export](#)
 - [SOAP / REST API](#)
 - [Sample Data](#)
- [Price Selection](#)
 - [Customer Group Overrides](#)
 - [Customer Overrides + Fallback](#)
 - [Formula Pricing](#)
 - [Sort Order Strategy](#)
 - [System Configuration \(Global\)](#)

Pricelist

Pricelist · /pricelist

Overview

The **Pricelist Extension** enables B2B merchants to create sophisticated pricing structures using named pricelists that act as containers for product prices. Instead of assigning prices directly to customers or groups, you create pricelists (e.g., "Wholesale 2025", "VIP Pricing", "US Region") and assign customers to these lists automatically based on customer or address attributes.

This architecture provides maximum flexibility for complex pricing scenarios and makes bulk pricing management significantly more efficient.

Key Features

Named Pricelist Management

Create pricelists as named containers with comprehensive metadata. Each pricelist can contain thousands of product prices organized in a single logical unit. Activate/deactivate entire pricing structures instantly or schedule future campaigns.

Automatic Customer Assignment

Define matching rules using customer and address attributes (group, company, tax ID, postcode, region, country) to automatically assign customers to appropriate pricelists. Eliminates manual assignments.

Customer Group Assignment

Assign pricelists to customer groups using the group assignment interface. All members of assigned groups automatically receive the pricelist. Supports both automated and manual pricing strategies.

Product Price Management

Store product prices within pricelists with support for quantity-based tiering (qty=1 standard, qty=10 bulk, qty=100 volume) and date scheduling for future price changes or promotions.

Priority-Based Resolution

Handle scenarios where customers match multiple pricelists using numeric priority values (0-999). Either select highest priority or merge quantity tiers from all matching pricelists.

Base Pricelist System

Designate one pricelist per website as "base pricelist" providing fallback pricing when customers don't match any specific pricelists. Ensures all customers see pricing.

What You Can Do

- Create named pricelists as containers
- Auto-assign customers based on attributes
- Assign pricelists to customer groups
- Create quantity-based pricing tiers
- Schedule time-based pricing campaigns
- Set different prices per website/region
- Merge pricing from multiple pricelists
- Designate base pricelist for fallback
- Import/export with add-on
- API access with add-on

Use Cases

Wholesale Tier Pricing

Create separate pricelists for wholesale tiers (Bronze, Silver, Gold) with automatic assignment based on purchase volume or company attributes.

Regional Pricing Management

Create pricelists for different regions (US, EU, APAC) with automatic assignment based on customer country or postcode.

Seasonal Campaign Pricing

Create time-limited pricelists for seasonal campaigns (Black Friday, Summer Sale) with date ranges that automatically activate and deactivate.

VIP Customer Programs

Create exclusive VIP pricelists with premium pricing and early access. Use highest priority to ensure VIP pricing always applies.

Contract-Based Pricing

Manage annual or multi-year customer contracts as named pricelists with precise date ranges matching contract terms.

Multi-Brand Pricing

Create separate pricelists for different brands or product lines. Assign customers based on authorized dealer status or purchase history.

Technical Details

Extension: MageB2B_Pricelist **Dependencies:** MageB2B_PricelistCore >= 2.3.0

Database Tables:

- `pricesystem_pricelist` - Pricelist containers
- `pricesystem_pricelist_group` - Group assignments

- `pricesystem_pricelist_product` - Product prices
- `pricesystem_pricelist_customer` - Customer assignments
- `pricesystem_pricelist_user` - User assignments

Price Code: pricelist

Getting Started

1. **Installation** - Install and configure the extension
2. **Creating Pricelists** - Add your first pricelist with products
3. **Named Pricelists** - Understanding pricelist containers
4. **Automatic Assignment** - Configure attribute-based matching
5. **Priority Resolution** - Handle multiple pricelist conflicts
6. **Price Selection** - How Pricesystem selects the final price
7. **My Prices** - Customer account price overview + CSV download

Add-Ons

Extend functionality with these add-ons:

- **SOAP / REST API Add-On** - CRUD WebAPI endpoints
- **CSV Import/Export Add-On** - Bulk import/export via CSV
- **Advanced Config Add-On** - Price selection overrides, custom sort order, formula pricing
- **Sample Data** - Demo entities and example prices

[View Extension Details](#)

Installation & Setup

Pricelist · [/pricelist/getting-started/installation](#)

Requirements

- Magento 2.4+
- PHP 8.0+
- MageB2B_PricesystemCore >= 2.3.0

Installation Steps

```
# Add authentication
composer config bearer.repo.softwaresilo.io <token>
composer config repositories.softwaresilo composer
https://repo.softwaresilo.io/

# Install
composer require mageb2b/pricesystem-pricelist:*

# Enable
php bin/magento module:enable MageB2B_PricesystemPricelist

# Setup
php bin/magento setup:upgrade
php bin/magento setup:di:compile
php bin/magento setup:static-content:deploy
php bin/magento cache:flush
```

Configuration

Navigate to: **Stores > Configuration > Pricesystem > Pricelist Settings**

Available Settings

Can Product Have Multi Qty Price

Controls whether products can have multiple quantity tiers within a pricelist.

- **Path:** `pricesystem/pricelist/can_product_have_multi_qty_price`
- **Type:** Yes/No
- **Default:** Yes
- **Effect:**
 - **Yes:** Products can have qty=1, qty=10, qty=50 tiers
 - **No:** Only one qty tier per product

Use Random Pricelist Name

Auto-generate unique pricelist names.

- **Path:** `pricesystem/pricelist/use_random_pricelistname`

- **Type:** Yes/No
- **Default:** No

Deactivate Old Pricelists

Automatically deactivate expired pricelists.

- **Path:** `pricesystem/pricelist/deactivate_old_pricelists`
- **Type:** Yes/No
- **Default:** No
- **Effect:** When enabled, pricelists with `to_date < today` are auto-deactivated

Enable Auto Assign Customers

Enable automatic customer assignment based on attributes.

- **Path:** `pricesystem/pricelist/enable_auto_assign_customers_to_pricelist`
- **Type:** Yes/No
- **Default:** Yes
- **Critical:** Must be enabled for attribute-based matching to work

Match Exact Type

Comparison mode for attribute matching.

- **Path:** `pricesystem/pricelist/match_exact_type`
- **Type:** Yes/No
- **Default:** No
- **Effect:**
 - **No (Loose):** "ACME" matches "ACME Corp" (contains)
 - **Yes (Exact):** "ACME" only matches "ACME" (strict equality)

Merge Pricelist Quantities

Combine quantity tiers from multiple pricelists.

- **Path:** `pricesystem/pricelist/merge_pricelist_qtys`
- **Type:** Yes/No
- **Default:** No
- **Effect:**
 - **No:** Use highest priority pricelist only
 - **Yes:** Merge qty tiers from all matching pricelists

Enable Base Pricelist Validation

Enforce only one base pricelist per website.

- **Path:** `pricesystem/pricelist/enable_base_pricelist_validation`
- **Type:** Yes/No
- **Default:** Yes
- **Recommended:** Keep enabled to prevent conflicts

Verification

```
php bin/magento module:status MageB2B_PricesystemPricelist
```

Expected: Module is enabled

Admin Access

Navigate to: **Catalog > Pricelists**

You should see:

- **Pricelists** - Main pricelist management grid
- Ability to create new pricelists
- Assign customers/groups
- Manage product prices

Database Tables

pricesystem_pricelist

Main pricelist container with metadata.

Key Fields:

- name - Pricelist name
- is_active - Active status
- priority - Conflict resolution priority
- from_date / to_date - Date range
- website_id - Website scope
- is_base_pricelist - Base pricelist flag
- Customer matching attributes (group, company, tax, postcode, region, country)

pricesystem_pricelist_group

Customer group assignments.

Key Fields:

- pricelist_id → pricesystem_pricelist
- group_id → customer_group

pricesystem_pricelist_product

Product prices within pricelists.

Key Fields:

- pricelist_id → pricesystem_pricelist
- product_id → catalog_product_entity
- qty - Quantity tier
- price - Price value
- from_date / to_date - Date range

pricesystem_pricelist_customer

Manual customer assignments (alternative to automatic).

Key Fields:

- pricelist_id → pricesystem_pricelist
- customer_id → customer_entity

pricesystem_pricelist_user

User assignments for admin-specific pricing.

Key Fields:

- pricelist_id → pricesystem_pricelist
- user_id → admin_user

Next Steps

- [Creating Your First Pricelist](#)
- [Named Pricelist Management](#)
- [Automatic Assignment](#)
- [Priority Resolution](#)

Creating Pricelists

Pricelist · </pricelist/getting-started/creating-pricelists>

Step-by-step guide to creating your first pricelist with products and customer assignments.

Quick Start

1. **Catalog > Pricelists** → Click "Add New"
 2. Fill basic info (name, dates, priority)
 3. Configure customer matching rules OR assign groups
 4. Save pricelist
 5. Add products with prices
 6. Activate pricelist
-

Creating a Pricelist Container

Step 1: Navigate to Grid

Admin Panel > Catalog > Pricelists

Grid shows existing pricelists with:

- Name
- Active status
- Priority
- Date range
- Website
- Base pricelist flag
- Actions

Step 2: Add New Pricelist

Click **Add New** button.

Step 3: Fill Basic Information

Name (Required)

Descriptive name for the pricelist.

Examples:

- "Wholesale 2025"
- "VIP Pricing Q1"
- "US Region Standard"
- "Black Friday Sale"
- "Contract - ACME Corp"

Is Active (Required)

Toggle pricelist on/off.

- **Yes:** Pricelist is evaluated for customer matching
- **No:** Pricelist ignored, even if customers match

Use Case: Prepare pricelists in advance, activate when ready.

Priority (Optional)

Numeric value (0-999) for conflict resolution.

- **Default:** 0
- **Higher = precedence**
- **Examples:**
 - 10: Standard pricing
 - 20: Promotional pricing
 - 30: VIP pricing
 - 40: Contract overrides

From Date (Optional)

Start date for pricelist validity.

- **Format:** YYYY-MM-DD
- **NULL:** Active immediately
- **Example:** 2025-06-01 (Summer campaign start)

To Date (Optional)

End date for pricelist validity.

- **Format:** YYYY-MM-DD
- **NULL:** No expiration
- **Must be >= From Date**
- **Example:** 2025-08-31 (Summer campaign end)

Website (Required)

Website scope for pricelist.

- **Default:** Default website
- **Multi-website:** Select specific site
- **Effect:** Only customers on this website see prices

Is Base Pricelist (Optional)

Designate as fallback pricelist.

- **Default:** No
- **Effect:** Customers without specific pricelist assignments get this pricelist
- **Limit:** Only ONE base pricelist per website
- **Validation:** Enforced by `enable_base_pricelist_validation` config

Customer Matching Attributes

These fields enable **automatic customer assignment**. When customers match attributes, they're auto-assigned to pricelist.

Customer Group

Match by customer group ID.

- **Example:** Group ID 2 (Wholesale)
- **Effect:** All wholesale customers get this pricelist

Company

Match by company name from customer account.

- **Example:** "ACME Corporation"
- **Loose matching:** "ACME" matches "ACME Corp" if `match_exact_type = No`
- **Exact matching:** Must be exactly "ACME" if `match_exact_type = Yes`

Tax/VAT Number

Match by tax ID.

- **Example:** "DE123456789"
- **Use Case:** EU VAT-registered businesses

Postcode

Match by billing/shipping postcode.

- **Example:** "90210"
- **Use Case:** Regional pricing by ZIP code

Region

Match by region/state.

- **Example:** "California"
- **Use Case:** State-specific pricing

Country

Match by country code.

- **Example:** "US"
- **Use Case:** Country-specific pricing

Note: Leave all matching fields empty to use manual assignment only (via groups or customer table).

Step 4: Save Pricelist

Click **Save** button.

Pricelist container created! Now add products.

Adding Products to Pricelist

Method 1: From Pricelist Edit Page

1. **Edit pricelist** → Navigate to "Products" tab
2. **Click "Add Products"**
3. **Select products** from grid (checkboxes)
4. **Set prices:**
 - Quantity: Tier threshold (e.g., 1, 10, 50)
 - Price: Price value
 - From Date: Start date (optional)
 - To Date: End date (optional)
5. **Save**

Method 2: Import/Export Add-On

Use the **CSV Import/Export Add-On** for bulk changes:

- [Pricelist CSV Import/Export Add-On](#)
- [Pricesystem CSV Import/Export \(overview\)](#)

The Magento ImportExport entity **Pricesystem Pricelist** supports importing/exporting pricelists including assignments and product rows.

Tip: Export first and use the exported CSV as your template (it matches your installed version/config).

Creating Quantity Tiers

Add multiple price entries for same product with different qty values.

Example: Graduated Wholesale Pricing

Pricelist: "Wholesale 2025" Product: "Widget Pro"

Qty Tier	Price	Description
1	\$100	Standard price
10	\$95	Bulk discount (5%)
50	\$90	Volume discount (10%)
100	\$85	Major volume (15%)

Customer orders 75 units: Gets \$90/unit (Qty 50 tier)

Configuration Required:

- `can_product_have_multi_qty_price = Yes`
-

Assigning Customers to Pricelists

Method 1: Automatic Assignment (Recommended)

Configure matching attributes on pricelist (see above).

Example:

- Pricelist: "Wholesale 2025"
- Customer Group: 2 (Wholesale)
- Region: "California"

Result: All wholesale customers in California auto-assigned.

Configuration Required:

- `enable_auto_assign_customers_to_pricelist = Yes`

Method 2: Group Assignment

Pricelist Edit Page > Groups Tab:

1. Click "Add Groups"
2. Select customer groups (Wholesale, Retail, VIP)
3. Save

Result: All members of selected groups assigned.

Method 3: Manual Customer Assignment

Pricelist Edit Page > Customers Tab:

1. Click "Add Customers"
2. Search and select customers
3. Save

Result: Only selected customers assigned.

Use Case: Override automatic rules for specific customers.

Common Workflows

Workflow 1: Wholesale Pricelist

Goal: Create wholesale pricing for 2025.

1. Create pricelist:

- Name: "Wholesale 2025"
- Active: Yes
- Priority: 15
- From Date: 2025-01-01
- To Date: 2025-12-31
- Customer Group: 2 (Wholesale)

2. Add products:

- Import CSV with all products
- Or add manually via Products tab

3. Test:

- Log in as wholesale customer
- Verify prices display

Workflow 2: Seasonal Campaign

Goal: Black Friday sale (Nov 29 - Dec 2, 2025).

1. Create pricelist:

- Name: "Black Friday 2025"
- Active: Yes
- Priority: 25 (higher than standard)
- From Date: 2025-11-29
- To Date: 2025-12-02
- Customer Group: 1 (General - all customers)

2. Add products:

- Select sale products
- Set discounted prices

3. Automatic:

- Activates Nov 29
- Deactivates Dec 3
- No manual intervention

Workflow 3: Regional Pricing

Goal: US customers get USD pricing.

1. Create pricelist:

- Name: "US Standard Pricing"

- Active: Yes
- Priority: 10
- Website: US Website
- Country: US

2. **Add products:**

- All products with USD prices

3. **Result:**

- US customers see USD prices
- EU customers see different pricelist (EUR)

Workflow 4: VIP Customer Program

Goal: Top customers get 15% off everything.

1. **Create pricelist:**

- Name: "VIP 15% Discount"
- Active: Yes
- Priority: 30 (highest)
- Customer Group: 4 (VIP)

2. **Add products:**

- All products with 15% discount applied
- Or use price calculation rule if available

3. **Result:**

- VIP customers automatically see reduced prices

Workflow 5: Contract Pricing

Goal: ACME Corp has annual contract with specific pricing.

1. **Create pricelist:**

- Name: "Contract - ACME Corp 2025"
- Active: Yes
- Priority: 30
- From Date: 2025-01-01
- To Date: 2025-12-31
- Company: "ACME Corporation" (exact match)

2. **Add products:**

- Contract products with agreed prices

3. **Renewal:**

- Create "Contract - ACME Corp 2026" before expiry
- Set From Date: 2026-01-01

- Seamless transition on Jan 1
-

Priority System in Action

Scenario: Customer Matches Multiple Pricelists

Setup:

- Customer: John Doe (Wholesale group, California, ACME Corp)
- Pricelist A: "Wholesale 2025" (Priority 15) - Matches group
- Pricelist B: "California Special" (Priority 20) - Matches region
- Pricelist C: "ACME Contract" (Priority 30) - Matches company

Configuration 1: merge_pricelist_qty = No

- **Result:** Only Pricelist C applies (highest priority)

Configuration 2: merge_pricelist_qty = Yes

- **Result:** Merge qty tiers from all three
 - Pricelist A: qty=1 (\$100), qty=10 (\$95)
 - Pricelist B: qty=25 (\$92)
 - Pricelist C: qty=50 (\$88)
 - **Customer gets:** qty=1 (\$100), qty=10 (\$95), qty=25 (\$92), qty=50 (\$88)
 - **Best of all lists!**
-

Base Pricelist Setup

Purpose

Fallback pricing for customers without specific pricelists.

Setup

1. Create pricelist:

- Name: "Base Pricing - Main Website"
- Active: Yes
- Priority: 0 (lowest)
- Is Base Pricelist: **Yes**
- Website: Main Website
- **Leave all matching attributes empty**

2. Add products:

- All products with default prices

3. Effect:

- Customers without any pricelist → see base prices
- Customers with specific pricelist → base ignored

Validation:

- Only ONE base pricelist per website
 - System prevents creating second base pricelist if validation enabled
-

Troubleshooting

Pricelist Not Showing for Customer

Check:

1. ✓ Is Active = Yes?
2. ✓ Date range valid? (from_date <= today <= to_date)
3. ✓ Website matches?
4. ✓ Customer matches attributes OR assigned via group/manual?
5. ✓ Auto-assignment enabled? (enable_auto_assign_customers_to_pricelist = Yes)
6. ✓ Cache cleared?

Wrong Prices Displaying

Check:

1. ✓ Multiple pricelists match? (check priority)
2. ✓ Merge config? (merge_pricelist_qtyys setting)
3. ✓ Product exists in pricelist?
4. ✓ Quantity tier matches cart quantity?
5. ✓ Pricesystem Core price order correct?

Can't Create Second Base Pricelist

Expected behavior!

- Only one base pricelist per website
 - Disable enable_base_pricelist_validation to override (not recommended)
 - Or deactivate existing base pricelist first
-

Next Steps

- [Named Pricelist Management](#)
- [Automatic Assignment](#)
- [Priority Resolution](#)

Automatic Customer Assignment

Pricelist · </pricelist/features/automatic-assignment>

Automatically assign customers to pricelists based on their attributes - no manual assignment needed.

Overview

Automatic assignment eliminates manual customer-to-pricelist mapping. Define matching rules on each pricelist using customer and address attributes. When customers log in or update profiles, the system evaluates all active pricelists and auto-assigns matching ones.

Benefits

- **Zero manual work:** Customers automatically get correct pricing
 - **Real-time updates:** Changes apply immediately on login
 - **New customer ready:** First-time customers get pricing instantly
 - **Attribute-driven:** Business logic encoded in rules
 - **Scalable:** Works with 10 or 10,000 customers
-

How It Works

Evaluation Trigger

System evaluates pricelists when:

1. Customer logs in
2. Customer updates profile/address
3. Customer group changes
4. Admin manually triggers re-evaluation
5. Cron job (if configured)

Evaluation Process

1. Get all ACTIVE pricelists for customer's website
2. Filter by date validity (`from_date <= today <= to_date`)
3. For each pricelist, check matching attributes:
 - Customer group matches?
 - Company name matches?
 - Tax ID matches?
 - Postcode matches?
 - Region matches?
 - Country matches?
4. If ALL filled attributes match → Assign customer to pricelist
5. Apply priority/merge logic for conflicts

Important: ALL filled attributes must match (AND logic, not OR).

Configuration

Enable Auto-Assignment

Path: Stores > Configuration > Pricesystem > Pricelist > Enable Auto Assign Customers

Setting: pricesystem/pricelist/enable_auto_assign_customers_to_pricelist

Values:

- **Yes (Required):** Auto-assignment enabled
- **No:** Only manual assignments work

Critical: Must be enabled for attribute-based matching!

Match Type

Path: Stores > Configuration > Pricesystem > Pricelist > Match Exact Type

Setting: pricesystem/pricelist/match_exact_type

Values:

Loose Matching (No - Default)

Partial string match, case-insensitive.

Examples:

- Pricelist company: "ACME"
- Customer company: "ACME Corporation" → **MATCH** (contains "ACME")
- Customer company: "AcMe Corp" → **MATCH** (case-insensitive)

Use Case: Flexible matching, forgives typos/variations.

Exact Matching (Yes)

Strict equality, case-sensitive.

Examples:

- Pricelist company: "ACME"
- Customer company: "ACME Corporation" → **NO MATCH** (not exact)
- Customer company: "ACME" → **MATCH** (exact)
- Customer company: "acme" → **NO MATCH** (case-sensitive)

Use Case: Precise matching, security-sensitive scenarios.

Matching Attributes

Customer Group ID

Field: `customer_group_id`

Source: Customer entity → `group_id` field

Example:

- Pricelist: `group_id = 2` (Wholesale)
- Customer: `group_id = 2`
- **Result:** MATCH

Use Cases:

- Wholesale/retail pricing
- VIP tiers
- Member-only pricing

Multi-Group: Customers belong to exactly ONE group at a time. Match is simple equality check.

Company Name

Field: `customer_company`

Source: Customer entity → `company` attribute (default or custom)

Example:

```
Pricelist: customer_company = "ACME Corporation"
Customer: company = "ACME Corporation"
Match: Yes (exact) or Yes (loose if contains "ACME")
```

Use Cases:

- B2B contract pricing
- Enterprise customers
- Corporate accounts

Best Practice:

- Use loose matching for flexibility
 - Standardize company names in CRM
 - Document approved variations
-

Tax/VAT Number

Field: `customer_tax`

Source: Customer entity → `taxvat` attribute

Example:

```
Pricelist: customer_tax = "DE123456789"  
Customer: taxvat = "DE123456789"  
Match: Yes
```

Use Cases:

- EU VAT-registered businesses
- Tax-exempt organizations
- Verified business accounts

Notes:

- Usually exact match (even with loose mode)
 - Format-sensitive (include dashes, spaces as stored)
 - Verify data quality
-

Postcode

Field: `address_postcode`

Source: Customer billing or shipping address → `postcode`

Example:

```
Pricelist: address_postcode = "90210"  
Customer billing: postcode = "90210"  
Match: Yes
```

Use Cases:

- ZIP code-based regional pricing
- Local delivery zones
- State-specific pricing

Notes:

- Checks both billing and shipping addresses
 - Match if either address matches
 - Loose mode: "9021" matches "90210"
-

Region

Field: `address_region`

Source: Customer address → `region` text field

Example:

```
Pricelist: address_region = "California"  
Customer billing: region = "California"  
Match: Yes
```

Use Cases:

- State/province pricing
- Regional campaigns
- Tax jurisdictions

Data Quality:

- Magento stores region as text (not ID)
 - "California" vs "CA" might not match
 - Standardize or use loose matching
-

Country

Field: address_country

Source: Customer address → country_id (2-letter ISO code)

Example:

```
Pricelist: address_country = "US"  
Customer billing: country_id = "US"  
Match: Yes
```

Use Cases:

- Country-specific pricing
- Currency localization
- Regulatory compliance

Notes:

- Always 2-letter ISO codes (US, DE, GB, FR, etc.)
 - Exact match even in loose mode
 - Most common matching attribute
-

Multi-Attribute Matching

AND Logic

All filled attributes must match simultaneously.

Example 1: Two Attributes

```
Pricelist:
- customer_group_id: 2 (Wholesale)
- address_country: US

Customer:
- group_id: 2 (Wholesale) ✓
- country: US ✓
Result: MATCH (both match)
```

Example 2: Partial Match Fails

```
Pricelist:
- customer_group_id: 2 (Wholesale)
- address_country: US

Customer:
- group_id: 2 (Wholesale) ✓
- country: DE ✗
Result: NO MATCH (country doesn't match)
```

Example 3: More Attributes

```
Pricelist:
- customer_group_id: 2
- customer_company: "ACME"
- address_country: US
- address_region: "California"

Customer:
- group_id: 2 ✓
- company: "ACME Corporation" ✓ (loose match)
- country: US ✓
- region: "California" ✓
Result: MATCH (all 4 match)
```

Empty Attributes Ignored

Null/empty attributes are not checked.

Example:

```
Pricelist:
- customer_group_id: 2
- address_country: NULL (not filled)

Customer:
- group_id: 2 ✓
- country: DE (doesn't matter, pricelist doesn't check country)
```

Result: MATCH (only group is checked)

Strategy: Leave attributes empty to broaden matching.

Common Scenarios

Scenario 1: Wholesale Customers Only

Goal: All wholesale customers get this pricelist.

Pricelist:

- customer_group_id: 2 (Wholesale)
- All other attributes: NULL

Result: Any customer in Wholesale group matches, regardless of location/company.

Scenario 2: US Wholesale Customers

Goal: Wholesale customers in US only.

Pricelist:

- customer_group_id: 2 (Wholesale)
- address_country: US
- All other attributes: NULL

Result: Must be wholesale AND in US.

Scenario 3: California Wholesale for ACME Corp

Goal: Very specific targeting.

Pricelist:

- customer_group_id: 2 (Wholesale)
- customer_company: "ACME"
- address_region: "California"
- address_country: US

Result: Only wholesale ACME employees in California, USA.

Scenario 4: EU VAT-Registered Businesses

Goal: EU businesses with VAT number.

Pricelist:

- customer_tax: "DE%" (if using wildcards) OR specific VAT
- address_country: DE/FR/IT (create multiple pricelists per country)

Result: VAT-registered businesses in target countries.

Note: Might need one pricelist per country or use group assignment instead.

Scenario 5: Regional Postcode Ranges

Goal: Specific ZIP codes get special pricing.

Challenge: Postcode ranges not directly supported.

Workaround 1: Multiple Pricelists

- Create pricelist for "90210"
- Create pricelist for "90211"
- Create pricelist for "90212"
- Assign same products/prices

Workaround 2: Custom Attribute

- Create custom customer attribute: "pricing_zone"
 - Set via import/automation based on postcode
 - Match on custom attribute instead
-

Manual Override

Group Assignment

Assign entire customer groups to pricelist, overriding attribute matching.

Path: Pricelist Edit > Groups Tab

Use Case: "All VIP members get this pricelist, regardless of attributes."

Combines with automatic: Customer can match via attributes AND be in assigned group.

Direct Customer Assignment

Assign specific customers to pricelist manually.

Path: Pricelist Edit > Customers Tab

Use Case: "Customer #12345 gets this contract pricelist (special exception)."

Overrides automatic: Manually assigned customers always get pricelist, even if attributes don't match.

Testing Automatic Assignment

Test Plan

1. **Create test pricelist:**

- Name: "Test - Wholesale US"
- customer_group_id: 2
- address_country: US
- Active: Yes

2. **Create test customer:**

- Group: Wholesale (2)
- Country: US
- Company: "Test Co"

3. **Test assignment:**

- Log in as test customer
- Check assigned pricelists (admin or API)
- Verify pricelist appears

4. **Test exclusion:**

- Change customer country to DE
- Log out, log back in
- Verify pricelist NOT assigned

5. **Test re-assignment:**

- Change customer back to US
 - Log out, log back in
 - Verify pricelist re-assigned
-

Troubleshooting

Customer Not Assigned to Pricelist

Checklist:

1. ✓ Auto-assignment enabled? (`enable_auto_assign_customers_to_pricelist = Yes`)

2. ✓ Pricelist active? (is_active = Yes)
3. ✓ Date range valid? (from_date <= today <= to_date)
4. ✓ Website matches? (customer on correct website)
5. ✓ All attributes match? (check each filled attribute)
6. ✓ Customer logged in recently? (triggers evaluation)
7. ✓ Cache cleared?

Debug:

```
-- Check customer attributes
SELECT entity_id, group_id, email FROM customer_entity WHERE entity_id = 123;

SELECT company, taxvat FROM customer_entity_varchar WHERE entity_id = 123
AND attribute_id IN (...);

-- Check pricelist matching attributes
SELECT * FROM pricesystem_pricelist WHERE id = 456;

-- Check assignments
SELECT * FROM pricesystem_pricelist_customer WHERE customer_id = 123;
```

Wrong Pricelist Assigned

Causes:

1. Multiple pricelists match (check priority)
2. Manual assignment overriding automatic
3. Group assignment in addition to automatic
4. Outdated cache

Resolution:

1. Review all pricelists customer matches
2. Check priority values
3. Clear customer pricelist assignments (if testing)
4. Re-login to trigger fresh evaluation

Loose vs Exact Matching Issues

Symptom: Customer matches when they shouldn't (or vice versa).

Check: match_exact_type configuration

Example:

- Pricelist company: "ACME"
- Customer company: "ACME Corporation"
- Loose: Matches
- Exact: Doesn't match

Fix: Choose appropriate mode or standardize data.

Best Practices

Data Quality

- Standardize company names
- Validate tax IDs on registration
- Use consistent region naming
- Encourage complete profiles

Attribute Selection

- Start with simple (group + country)
- Add attributes only when needed
- Document matching logic
- Test edge cases

Performance

- Don't create too many pricelists (100s are fine, 10,000s might be slow)
- Use group assignment for large segments
- Consider caching strategies

Documentation

- Document which attributes each pricelist uses
- Explain business rules to team
- Maintain matching logic reference

Related Documentation

- [Creating Pricelists](#)
- [Named Pricelists](#)
- [Priority Resolution](#)

Named Pricelist Management

Pricelist · </pricelist/features/named-pricelists>

Organize product prices into named containers that act as logical pricing units.

Overview

Named pricelists are **containers** that group product prices together under a descriptive name. Instead of assigning individual prices to customers/groups, you create pricelists like "Wholesale 2025" or "VIP Tier" and manage entire pricing structures as units.

Key Advantages

1. Logical Organization

- Group prices by business logic (season, tier, region, contract)
- Easy to find and manage related prices
- Clear naming conventions

2. Bulk Operations

- Activate/deactivate entire pricing structures instantly
- One toggle affects thousands of products
- Schedule future campaigns

3. Date-Based Control

- Set from_date/to_date on container
- Automatic activation/deactivation
- No manual intervention

4. Priority Management

- Each pricelist has priority value
- Control which pricing wins conflicts
- Fine-grained resolution

Pricelist Container Fields

Name (Required)

Descriptive identifier for the pricelist.

Naming Conventions:

- **Temporal:** "Q1 2025 Pricing", "Summer Sale 2025"
- **Segment:** "Wholesale Bronze", "VIP Gold Tier"
- **Regional:** "US Standard", "EU VAT Pricing"
- **Contract:** "Contract - ACME Corp 2025"
- **Campaign:** "Black Friday 2025", "Flash Sale #42"

Best Practices:

- Include year for time-bound pricing
- Include segment/tier for clarity
- Avoid generic names ("Test", "New", "Pricing")

Is Active (Boolean)

Toggle pricelist on/off without deleting.

Use Cases:

- **Inactive during setup:** Prepare pricelist, activate when ready
- **Temporary disable:** Turn off problematic pricelist quickly
- **Seasonal rotation:** Deactivate winter pricing, activate summer

Effect:

- Active = Yes: Pricelist evaluated for customer matching
- Active = No: Pricelist completely ignored

Priority (0-999)

Numeric value for conflict resolution.

Default: 0 **Range:** 0-999 **Rule:** Higher priority wins

Recommended Hierarchy:

- 0-9: Base pricelists
- 10-19: Standard pricelists
- 20-29: Promotional pricelists
- 30-39: VIP/Contract pricelists
- 40+: Emergency overrides

Example:

- "Standard Wholesale" (Priority 15)
- "Summer Campaign" (Priority 25) → Overrides standard during summer
- "VIP Contract" (Priority 35) → Always wins for VIP customers

From Date / To Date

Date range for automatic activation/deactivation.

Format: YYYY-MM-DD or NULL

Scenarios:

Permanent (Both NULL):

```
from_date: NULL
to_date: NULL
```

Active forever (or until manually deactivated).

Future Start (From Date set):

```
from_date: 2025-06-01  
to_date: NULL
```

Activates June 1st, runs forever.

Fixed Campaign (Both set):

```
from_date: 2025-11-29  
to_date: 2025-12-02
```

Black Friday campaign, auto-expires after Cyber Monday.

Grace Period:

```
from_date: 2025-01-01  
to_date: 2026-01-31
```

Annual contract + 1 month grace period for renewal.

Website ID

Website scope for multi-website installations.

Options:

- Specific website (US Website, EU Website)
- All websites (website_id = 0)

Use Case:

- US Website: USD pricing
- EU Website: EUR pricing with VAT
- UK Website: GBP pricing

Is Base Pricelist (Boolean)

Designate as fallback pricelist for website.

Rules:

- Only ONE base pricelist per website
- Enforced by `enable_base_pricelist_validation`
- Lowest priority (usually 0)
- No matching attributes (applies to everyone without specific pricelist)

Purpose:

- Ensure all customers see some pricing
- Prevent "no price" situations
- Fallback for new customers

Customer Matching Attributes

These fields enable automatic customer assignment.

Customer Group ID

Match by Magento customer group.

Example:

- Group 2 = Wholesale
- All wholesale customers get this pricelist

Company Name

Match by company field in customer account.

Example:

- Company: "ACME Corporation"
- All ACME employees get contract pricing

Matching Mode:

- Loose (`match_exact_type = No`): "ACME" matches "ACME Corp"
- Exact (`match_exact_type = Yes`): Must be exactly "ACME"

Tax/VAT Number

Match by tax identification number.

Example:

- Tax: "DE123456789"
- German VAT-registered businesses

Postcode

Match by billing/shipping postcode.

Example:

- Postcode: "90210"
- Beverly Hills residents

Region

Match by state/province/region.

Example:

- Region: "California"

- California-specific pricing

Country

Match by country code.

Example:

- Country: "US"
- United States pricing

Multi-Attribute Matching:

All filled attributes must match (AND logic).

Example:

```
Customer Group: Wholesale (2)
Country: US
Region: California
```

Only wholesale customers in California, USA match.

Database Structure

Table: pricesystem_pricelist

```
CREATE TABLE pricesystem_pricelist (
  id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(255) NOT NULL,
  is_active TINYINT DEFAULT 1,
  priority INT DEFAULT 0,
  from_date DATE DEFAULT NULL,
  to_date DATE DEFAULT NULL,
  website_id INT NOT NULL,
  is_base_pricelist TINYINT DEFAULT 0,

  -- Customer matching attributes
  customer_group_id INT DEFAULT NULL,
  customer_company VARCHAR(255) DEFAULT NULL,
  customer_tax VARCHAR(255) DEFAULT NULL,
  address_postcode VARCHAR(255) DEFAULT NULL,
  address_region VARCHAR(255) DEFAULT NULL,
  address_country VARCHAR(2) DEFAULT NULL,

  created_at TIMESTAMP,
  updated_at TIMESTAMP,

  FOREIGN KEY (website_id) REFERENCES store_website(website_id),
```

```
FOREIGN KEY (customer_group_id) REFERENCES
customer_group(customer_group_id)
);
```

Admin Interface

Location

Admin Panel > Catalog > Pricelists

Grid Columns

- ID
- Name
- Active (Yes/No)
- Priority
- From Date
- To Date
- Website
- Base Pricelist (Yes/No)
- Created
- Actions (Edit/Delete)

Grid Features

- Filter by name, active status, website
- Sort by any column
- Mass activate/deactivate
- Export to CSV (with add-on)

Edit Form

Tabs:

1. **General** - Name, active, priority, dates, website, matching attributes
 2. **Products** - Add/manage product prices within pricelist
 3. **Groups** - Assign customer groups
 4. **Customers** - Manual customer assignments
 5. **Users** - Admin user assignments
-

Common Workflows

Workflow 1: Annual Pricing Cycle

Goal: Replace 2024 pricing with 2025 pricing on Jan 1.

Setup:

1. **Existing:** "Wholesale 2024" (Priority 15, from_date: 2024-01-01, to_date: 2024-12-31)
2. **New:** "Wholesale 2025" (Priority 15, from_date: 2025-01-01, to_date: 2025-12-31)
3. **Result:** Auto-switch on January 1st, no downtime

Workflow 2: Promotional Override

Goal: Black Friday sale overrides standard pricing.

Setup:

1. **Standard:** "Wholesale 2025" (Priority 15, permanent)
2. **Campaign:** "Black Friday 2025" (Priority 25, from: 2025-11-29, to: 2025-12-02)
3. **Result:** Higher priority campaign wins during Black Friday, then reverts to standard

Workflow 3: Regional Expansion

Goal: Launch EU pricing alongside US pricing.

Setup:

1. **US:** "US Standard Pricing" (website_id: US, country: US)
2. **EU:** "EU Standard Pricing" (website_id: EU, country: DE/FR/IT/etc)
3. **Result:** Automatic regional pricing based on customer country

Workflow 4: VIP Program

Goal: VIP customers always get best pricing.

Setup:

1. **Standard:** Various pricelists (Priority 10-20)
2. **VIP:** "VIP Exclusive" (Priority 35, group_id: VIP)
3. **Result:** VIP pricelist always wins due to highest priority

Best Practices

Naming

- "Wholesale Q1 2025" - Clear, dated
- "Contract - ACME Corp 2025-2026" - Contract with dates
- "US Region Standard Pricing" - Region + tier
- "Test" - Too generic
- "Pricing 2" - No context
- "asdf" - Not descriptive

Priority Values

- Use consistent tiers (0, 10, 20, 30)
- Leave gaps (10, 15, 20 not 10, 11, 12)
- Document your hierarchy
- Avoid priority 999 (emergency only)

Date Ranges

- Include grace periods for renewals
- Overlap dates deliberately (e.g., campaign + standard)
- Don't leave gaps between annual cycles
- Test date transitions in staging

Activation

- Prepare inactive pricelists in advance
 - Test before activating
 - Activate during low-traffic periods
 - Monitor after activation
-

Troubleshooting

Pricelist Not Applying

Check:

1. Is Active = Yes?
2. Date range valid? (from <= today <= to)
3. Website matches customer's website?
4. Matching attributes correct?
5. Customer actually matches attributes?
6. Cache cleared?

Multiple Pricelists Conflict

Expected! Use priority system.

Check:

1. Priority values set correctly?
2. Merge config (merge_pricelist_qtys) set as desired?
3. Date ranges overlap intentionally?

Can't Create Base Pricelist

Cause: Base pricelist already exists for website.

Resolution:

1. Find existing base pricelist for website
 2. Either deactivate it or set is_base_pricelist = No
 3. Then create new base pricelist
-

Related Documentation

- [Creating Pricelists](#)
- [Automatic Assignment](#)
- [Priority Resolution](#)

Priority-Based Resolution

Pricelist · /pricelist/features/priority-resolution

Handle scenarios where customers match multiple pricelists using numeric priority values and merge strategies.

Overview

Customers can match multiple pricelists through:

- Different matching attributes (group + region + company)
- Manual group assignments
- Direct customer assignments
- Overlapping date ranges

The Priority System resolves these conflicts predictably.

Priority Field

Every pricelist has a `priority` field:

- **Type:** Integer
- **Range:** 0-999
- **Default:** 0
- **Rule:** Higher value = higher precedence

Example:

- Pricelist A: Priority 10
 - Pricelist B: Priority 20
 - **Winner:** Pricelist B
-

Configuration

Merge Pricelist Quantities

Path: Stores > Configuration > Pricesystem > Pricelist > Merge Pricelist Qtys

Setting: `pricesystem/pricelist/merge_pricelist_qtys`

Values:

No (Default): Select Highest Priority

When customer matches multiple pricelists, use ONLY the highest priority pricelist.

Example:

Customer matches:

```
- Pricelist A (Priority 10): Product X @ qty=1 ($100), qty=10 ($95)
- Pricelist B (Priority 20): Product X @ qty=1 ($98), qty=50 ($90)

Result: Use ONLY Pricelist B (higher priority)
- qty=1: $98
- qty=50: $90
(Pricelist A ignored completely)
```

Use Case: Simple, predictable pricing. One pricelist wins.

Yes: Merge Quantity Tiers

Combine quantity tiers from ALL matching pricelists, taking best price at each tier.

Example:

```
Customer matches:
- Pricelist A (Priority 10): Product X @ qty=1 ($100), qty=10 ($95)
- Pricelist B (Priority 20): Product X @ qty=1 ($98), qty=50 ($90)

Result: MERGE tiers, best price wins:
- qty=1: $98 (from Pricelist B, better than A's $100)
- qty=10: $95 (from Pricelist A, B has no qty=10)
- qty=50: $90 (from Pricelist B, A has no qty=50)

Customer gets: qty=1 ($98), qty=10 ($95), qty=50 ($90)
Best of both pricelists!
```

Use Case: Complex pricing scenarios, best-of-all-lists strategy.

Priority Hierarchy Recommendations

Standard Hierarchy

0-9: Base/Fallback

- Priority 0: Base pricelist (website default)
- Priority 5: Legacy/migration pricing

10-19: Standard Pricing

- Priority 10: Standard retail
- Priority 12: Standard wholesale
- Priority 15: Regional standard

20-29: Promotional Pricing

- Priority 20: Standard promotions
- Priority 22: Seasonal sales
- Priority 25: Flash sales

30-39: VIP/Contract

- Priority 30: VIP tier pricing
- Priority 32: Annual contracts
- Priority 35: Enterprise agreements

40-49: Emergency Overrides

- Priority 40: Temporary fixes
- Priority 45: Urgent campaigns

50+: Reserved

- Priority 50+: System overrides
 - Priority 99: Testing
 - Priority 999: Absolute override (avoid)
-

Resolution Scenarios

Scenario 1: Same Priority

Setup:

```
Pricelist A: Priority 10  
Pricelist B: Priority 10
```

Merge = No:

- **Undefined behavior!** (might be ID order, creation order, or random)
- **Solution:** Assign unique priorities (10 vs 11)

Merge = Yes:

- Merges tiers from both
- Best price wins at each tier
- Defined behavior

Best Practice: Avoid same-priority conflicts by using unique values.

Scenario 2: Clear Winner

Setup:

```
Pricelist A: Priority 10 (Standard wholesale)  
Pricelist B: Priority 30 (VIP contract)
```

Merge = No:

- Use Pricelist B only (Priority 30 > 10)

Merge = Yes:

- Merge tiers from both
 - VIP contract prices likely win at each tier (better pricing)
 - But if wholesale has unique tiers, customer gets those too
-

Scenario 3: Three-Way Conflict

Setup:

Customer matches three pricelists:

- Pricelist A: Priority 15 (Wholesale)
- Pricelist B: Priority 20 (California Regional)
- Pricelist C: Priority 30 (ACME Contract)

Product X prices:

- A: qty=1 (\$100), qty=10 (\$95)
- B: qty=1 (\$98), qty=25 (\$92)
- C: qty=1 (\$96), qty=50 (\$88)

Merge = No:

- Use ONLY Pricelist C (Priority 30)
- Customer gets: qty=1 (\$96), qty=50 (\$88)

Merge = Yes:

- Merge all three pricelists
- Best price at each tier:
 - qty=1: \$96 (from C)
 - qty=10: \$95 (from A)
 - qty=25: \$92 (from B)
 - qty=50: \$88 (from C)
- Customer gets: qty=1 (\$96), qty=10 (\$95), qty=25 (\$92), qty=50 (\$88)

Analysis: Merge gives best pricing across all tiers.

Scenario 4: Date-Based Priority Shift

Setup:

Standard pricing:

- Pricelist: "Wholesale 2025"
- Priority: 15
- Dates: 2025-01-01 to 2025-12-31
- Product X: qty=1 (\$100)

Campaign:

- Pricelist: "Black Friday 2025"
- Priority: 25
- Dates: 2025-11-29 to 2025-12-02
- Product X: qty=1 (\$75)

Timeline:

Nov 28:

- Only Wholesale active: \$100

Nov 29-30 (Black Friday):

- Both active, campaign wins (Priority 25 > 15): \$75

Dec 3+:

- Only Wholesale active again: \$100

Key Point: Temporary high-priority campaigns override standard pricing.

Scenario 5: Product Not in High-Priority List

Setup:

Pricelist A (Priority 10): Products X, Y, Z
Pricelist B (Priority 20): Products X, Y (Z missing)

Customer views Product Z:

Merge = No:

- Use Pricelist B (higher priority)
- Product Z not in Pricelist B
- **Result:** No pricelist price (fallback to catalog price)

Merge = Yes:

- Check all pricelists
- Product Z found in Pricelist A
- **Result:** Uses Pricelist A price for Product Z

Lesson: Merge mode more forgiving when high-priority lists incomplete.

Priority Management

Setting Priorities

At Creation:

1. Identify purpose (standard/promo/VIP/contract)
2. Check existing pricelists in same tier
3. Assign priority within tier
4. Document reasoning

Example:

```
Creating "Summer Sale 2025"  
Purpose: Seasonal promotion  
Tier: 20-29 (promotional)  
Existing: Spring Sale (22)  
Assign: 23 (next in sequence)  
Document: "Summer campaign, overrides standard"
```

Changing Priorities

Caution: Affects all customers assigned to pricelist.

Process:

1. Understand current behavior
2. Test priority change in staging
3. Verify customer impact
4. Update during low-traffic period
5. Monitor after change
6. Document change reason

Bulk Priority Updates

Scenario: Increase all VIP priorities by 10.

SQL:

```
UPDATE pricesystem_pricelist  
SET priority = priority + 10  
WHERE name LIKE 'VIP%'  
OR customer_group_id = 4; -- VIP group
```

Caution: Test first! Affects all customers.

Testing Priority Logic

Test Case 1: Basic Priority

Setup:

1. Create Pricelist A (Priority 10) with Product X = \$100
2. Create Pricelist B (Priority 20) with Product X = \$90
3. Assign test customer to both pricelists

Test Merge = No:

- Expected: \$90 (Pricelist B wins)
- Verify: Log in, check Product X price

Test Merge = Yes:

- Expected: \$90 (still \$90, both have same tier)
- Verify: Same result

Test Case 2: Merge Benefit

Setup:

1. Pricelist A (Priority 10): Product X @ qty=1 (\$100), qty=10 (\$95)
2. Pricelist B (Priority 20): Product X @ qty=1 (\$98), qty=50 (\$90)
3. Assign test customer to both

Test Merge = No:

- Expected: Only qty=1 (\$98), qty=50 (\$90) from Pricelist B
- Verify: qty=10 tier NOT available

Test Merge = Yes:

- Expected: qty=1 (\$98), qty=10 (\$95), qty=50 (\$90)
- Verify: qty=10 tier available (from Pricelist A)

Best Practices

1. Use Tiers, Not Random Values

Good:

- 0, 10, 20, 30, 40 (clear tiers)
- 10, 15, 20, 25, 30 (with gaps)

Bad:

- 7, 13, 21, 29, 33 (no pattern)
- 1, 2, 3, 4, 5 (no gaps)

2. Leave Gaps

Allows inserting priorities later without renumbering.

Example:

- Current: 10, 20, 30

- Need to insert between 10 and 20: Use 15
- With consecutive (10, 11, 12): Must renumber everything

3. Document Hierarchy

Create internal documentation:

Priority Hierarchy - MageB2B Pricelists

0-9: Base pricelists

0 = Website default pricing

10-19: Standard pricing

10 = Standard retail

12 = Standard wholesale

15 = Regional standard

20-29: Promotional pricing

20 = Standard promotions

25 = Flash sales/urgent campaigns

30-39: VIP/Contract pricing

30 = VIP tier

35 = Enterprise contracts

40+: Emergency/testing

4. Test Merge Configuration

Before enabling merge:

1. Test with 2-3 pricelists
2. Verify expected behavior
3. Check edge cases (missing products, same tiers)
4. Monitor after enabling

5. Avoid Priority 999

Reserve for absolute emergencies only. Creates maintenance issues.

6. Audit Regularly

Quarterly review:

- List pricelists by priority
- Find duplicates (same priority)
- Find gaps (orphaned tiers)
- Verify hierarchy still makes sense

Troubleshooting

Wrong Price Displaying

Checklist:

1. ✓ Which pricelists does customer match?
2. ✓ What are their priorities?
3. ✓ Is merge enabled?
4. ✓ Does product exist in high-priority list?
5. ✓ Are dates valid for all pricelists?
6. ✓ Cache cleared?

Debug SQL:

```
-- Find customer's pricelists
SELECT
    pl.id,
    pl.name,
    pl.priority,
    pl.is_active,
    pl.from_date,
    pl.to_date
FROM pricesystem_pricelist pl
JOIN pricesystem_pricelist_customer plc ON plc.pricelist_id = pl.id
WHERE plc.customer_id = 123
ORDER BY pl.priority DESC;

-- Check product prices in those pricelists
SELECT
    pl.name,
    pl.priority,
    plp.qty,
    plp.price
FROM pricesystem_pricelist_product plp
JOIN pricesystem_pricelist pl ON pl.id = plp.pricelist_id
WHERE plp.product_id = 456
AND plp.pricelist_id IN (SELECT pricelist_id FROM
pricesystem_pricelist_customer WHERE customer_id = 123)
ORDER BY pl.priority DESC, plp.qty ASC;
```

Merge Not Working

Symptoms: Customer only sees one pricelist's prices, not merged.

Checks:

1. ✓ `merge_pricelist_qty` = Yes in config?
2. ✓ Config cache cleared?
3. ✓ Customer actually assigned to multiple pricelists?
4. ✓ Pricelists have different qty tiers to merge?

Same Priority Issues

Symptom: Unpredictable pricing behavior.

Cause: Multiple pricelists with same priority.

Resolution:

1. Find duplicates: `SELECT priority, COUNT(*) FROM pricesystem_pricelist GROUP BY priority HAVING COUNT(*) > 1;`
 2. Assign unique priorities
 3. Test again
-

Advanced: Manual Priority Override

For specific customers, override priority system:

Method 1: Highest Priority Pricelist

Create customer-specific pricelist with priority 50+.

Example:

- Standard pricelists: 10-30
- Customer exception: Priority 60
- Always wins for that customer

Method 2: Date Range Stacking

Layer pricelists with different date ranges but same customer.

Example:

```
Base: Priority 10, permanent
Q1 Special: Priority 20, Jan-Mar
Q2 Special: Priority 20, Apr-Jun
VIP Override: Priority 30, permanent
```

Customer gets VIP pricing except during Q1/Q2 when specials might offer better prices.

Related Documentation

- [Creating Pricelists](#)
- [Named Pricelists](#)
- [Automatic Assignment](#)

Advanced Config

Pricelist · /pricelist/addons/advanced-config

The Advanced Config add-on extends Pricesystem with additional configuration and override options that affect how **pricelist prices** are selected and applied.

Package

- Composer package: `mageb2b/pricesystem-advancedconfig`
- Magento module: `MageB2B_PricesystemCoreAdvancedConfig`

Related

- [Price Selection](#)
- [Pricesystem Advanced Config](#)

CSV Import/Export

Pricelist · /pricelist/addons/csv-import-export

Bulk import and export **pricelists** and their assignments/products via CSV.

Package

- Composer package: `mageb2b/pricesystem-pricelist-importexport`
- Magento module: `MageB2B_PricesystemPricelistImportExport`

Installation

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

php bin/magento setup:upgrade
php bin/magento cache:flush
```

Admin Import/Export (Magento ImportExport)

1. Go to **System > Data Transfer > Import**
2. Entity type: **Pricesystem Pricelist**

Tip: Export first and use the exported CSV as your template.

CLI Commands

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

Related

- [Price Selection](#)
- [Pricesystem CSV Import/Export](#)

SOAP / REST API

Pricelist · /pricelist/addons/rest-api

Adds CRUD WebAPI endpoints for **pricelists** (pricelists, pricelist products, customer assignments, group assignments).

Package

- Composer package: `mageb2b/pricesystem-pricelist-api`
- Magento module: `MageB2B_PricesystemPricelistApi`

Installation

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

php bin/magento setup:upgrade
php bin/magento cache:flush
```

Endpoints (Overview)

Pricelist endpoints:

Method	URL
GET	/V1/pricesystem/pricelist/:id
GET	/V1/pricesystem/pricelist/search
POST	/V1/pricesystem/pricelist
PUT	/V1/pricesystem/pricelist/:id
POST	/V1/pricesystem/pricelistbulk
DELETE	/V1/pricesystem/pricelist/:id
DELETE	/V1/pricesystem/pricelist/all

Pricelist product endpoints:

Method	URL
GET	/V1/pricesystem/pricelistproduct/:id
GET	/V1/pricesystem/pricelistproduct/search
POST	/V1/pricesystem/pricelistproduct
POST	/V1/pricesystem/pricelistproductbulk
PUT	/V1/pricesystem/pricelistproduct/:id
PUT	/V1/pricesystem/pricelistproductbulk

Method	URL
DELETE	/V1/pricesystem/pricelistproduct/:id

Customer assignment endpoints:

Method	URL
GET	/V1/pricesystem/pricelistcustomer/:id
GET	/V1/pricesystem/pricelistcustomer/search
POST	/V1/pricesystem/pricelistcustomer
PUT	/V1/pricesystem/pricelistcustomer/:id
DELETE	/V1/pricesystem/pricelistcustomer/:id
DELETE	/V1/pricesystem/pricelistcustomer/deleteByCustomerId/:customerId

Customer group assignment endpoints:

Method	URL
GET	/V1/pricesystem/pricelistgroup/:id
GET	/V1/pricesystem/pricelistgroup/search
POST	/V1/pricesystem/pricelistgroup
PUT	/V1/pricesystem/pricelistgroup/:id
DELETE	/V1/pricesystem/pricelistgroup/:id

Authentication

All endpoints above require a valid admin bearer token.

```
curl -X POST "https://your-store.com/rest/V1/integration/admin/token" \
-H "Content-Type: application/json" \
-d '{"username":"admin","password":"***"}'
```

Then call endpoints with:

```
-H "Authorization: Bearer <ADMIN_TOKEN>"
```

Request Payloads

The following payloads are the request wrappers expected by Magento WebAPI for this module.

Pricelist Requests

GET /V1/pricesystem/pricelist/:id

No request body.

GET /V1/pricesystem/pricelist/search

No request body. Use `searchCriteria` query parameters, for example:

```
/V1/pricesystem/pricelist/search?searchCriteria[filter_groups][0][filters][0][field]=name&searchCriteria[filter_groups][0][filters][0][value]=B2B&searchCriteria[filter_groups][0][filters][0][condition_type]=like&searchCriteria[currentPage]=1&searchCriteria[pageSize]=20
```

POST /V1/pricesystem/pricelist and PUT /V1/pricesystem/pricelist/:id use wrapper key pricelist:

```
{
  "pricelist": {
    "id": 12,
    "name": "B2B Winter 2026",
    "active": true,
    "priority": 10,
    "from_date": "2026-03-01",
    "to_date": "2026-12-31",
    "website_id": 1,
    "products": [
      {
        "product_id": "24-MB01",
        "qty": 1,
        "price": "49.90",
        "price_application_type": 1,
        "start": "2026-03-01",
        "end": "2026-12-31",
        "price_attribute_id": 77
      }
    ],
    "customers": "42,99",
    "groups": "1,3",
    "is_base_pricelist": false,
    "customer_attribute": "",
    "customer_attribute_value": "",
    "customer_address_attribute": "",
    "customer_address_attribute_value": "",
    "customer_attribute_combine": false
  }
}
```

POST /V1/pricesystem/pricelistbulk uses wrapper key pricelists:

```
{
  "pricelists": [
    {
      "name": "PL A",
      "active": true,
      "priority": 10,
      "website_id": 1
    },
    {
      "id": 12,
```

```

    "name": "PL B Updated",
    "active": true,
    "priority": 20,
    "website_id": 1
  }
]
}

```

For the `pricelist` object, `customers` and `groups` are REST input strings (comma-separated), not arrays. For assignment workflows, prefer the dedicated endpoints:

`/V1/pricesystem/pricelistcustomer*` and `/V1/pricesystem/pricelistgroup*`.

DELETE `/V1/pricesystem/pricelist/:id` and DELETE `/V1/pricesystem/pricelist/all`
No request body.

Pricelist Product Requests

GET `/V1/pricesystem/pricelistproduct/:id`
No request body.

GET `/V1/pricesystem/pricelistproduct/search`
No request body. Example:

```

/V1/pricesystem/pricelistproduct/search?searchCriteria[filter_groups][0][filters][0][field]=pricelist_id&searchCriteria[filter_groups][0][filters][0][value]=12&searchCriteria[filter_groups][0][filters][0][condition_type]=eq

```

POST `/V1/pricesystem/pricelistproduct` and PUT
`/V1/pricesystem/pricelistproduct/:id` use wrapper key `pricelistProduct`:

```

{
  "pricelistProduct": {
    "id": 55,
    "pricelist_id": 12,
    "product_id": "24-MB01",
    "qty": 1,
    "price": "49.90",
    "price_application_type": 1,
    "start": "2026-03-01",
    "end": "2026-12-31",
    "price_attribute_id": 77
  }
}

```

POST `/V1/pricesystem/pricelistproductbulk` and PUT
`/V1/pricesystem/pricelistproductbulk` use wrapper key `pricelistProducts`:

```

{
  "pricelistProducts": [
    {
      "id": 55,

```

```

    "pricelist_id": 12,
    "product_id": "24-MB01",
    "qty": 1,
    "price": "47.90",
    "price_application_type": 1
  },
  {
    "id": 56,
    "pricelist_id": 12,
    "product_id": "24-MB04",
    "qty": 5,
    "price": "37.90",
    "price_application_type": 5
  }
]
}

```

DELETE /V1/pricesystem/pricelistproduct/:id
No request body.

Pricelist Customer Assignment Requests

GET /V1/pricesystem/pricelistcustomer/:id
No request body.

GET /V1/pricesystem/pricelistcustomer/search
No request body. Example:

```

/V1/pricesystem/pricelistcustomer/search?searchCriteria[filter_groups][0][filters][0][field]=pricelist_id&searchCriteria[filter_groups][0][filters][0][value]=12&searchCriteria[filter_groups][0][filters][0][condition_type]=eq

```

POST /V1/pricesystem/pricelistcustomer and PUT
/V1/pricesystem/pricelistcustomer/:id use wrapper key pricelistCustomer:

```

{
  "pricelistCustomer": {
    "id": 90,
    "pricelist_id": 12,
    "customer_id": "alice@example.com"
  }
}

```

DELETE /V1/pricesystem/pricelistcustomer/:id and DELETE
/V1/pricesystem/pricelistcustomer/deleteByCustomerId/:customerId
No request body.

Pricelist Group Assignment Requests

GET /V1/pricesystem/pricelistgroup/:id
No request body.

GET /V1/pricesystem/pricelistgroup/search

No request body. Example:

```
/V1/pricesystem/pricelistgroup/search?searchCriteria[filter_groups][0][filters][0][field]=pricelist_id&searchCriteria[filter_groups][0][filters][0][value]=12&searchCriteria[filter_groups][0][filters][0][condition_type]=eq
```

POST /V1/pricesystem/pricelistgroup and PUT /V1/pricesystem/pricelistgroup/:id
use wrapper key `pricelistGroup`:

```
{
  "pricelistGroup": {
    "id": 77,
    "pricelist_id": 12,
    "group_id": 3
  }
}
```

DELETE /V1/pricesystem/pricelistgroup/:id

No request body.

Field Notes

- `price_application_type` values:
 - 1 = fixed
 - 2 = surcharge nominal
 - 3 = surcharge percent
 - 4 = discount nominal
 - 5 = discount percent
- `product_id` can resolve from ID or SKU, depending on `pricesystem/api/product_attribute_code`.
- `customer_id` can resolve from ID or email, depending on `pricesystem/api/customer_attribute_code`.
- `website_id` supports numeric website ID and website code.

For the full list, see: [Pricesystem SOAP / REST API](#)

Sample Data

Pricelist · /pricelist/addons/sample-data

Installs demo entities and example **pricelist** records.

Packages

- Core sample data: mageb2b/pricesystem-sample-data
(MageB2B_PricesystemCoreSampleData)
- Pricelist sample data: mageb2b/pricesystem-pricelist-sample-data
(MageB2B_PricesystemPricelistSampleData)

Installation

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

php bin/magento setup:upgrade
php bin/magento cache:flush
```

Related

- [Pricesystem Sample Data](#)

Price Selection

Pricelist · /pricelist/price-selection

Pricelist is a Pricesystem price type.

Pricesystem selects the final purchasable price using a global strategy (Lowest / Highest / Sort order / Formula), and the Pricelist module also has internal selection logic (priority, base pricelist, merge qty tiers).

Quick Links

- [System Configuration \(Global\)](#)
- [Sort Order Strategy \(Priority List\)](#)
- [Customer Group Overrides](#)
- [Customer Overrides + Fallback](#)
- [Formula Pricing](#)

Pricelist In The Strategy

Price codes:

- `pricelist` (regular pricelist)
- `base_pricelist` (base pricelist fallback)

If you want pricelist prices to win, put `pricelist` (and optionally `base_pricelist`) near the top of your sort order list.

Customer Group Overrides

Pricelist · </pricelist/price-selection/customer-group-overrides>

Customer/group override strategies require:

- `mageb2b/pricesystem-advancedconfig` (`MageB2B_PricesystemCoreAdvancedConfig`)

When Group Overrides Apply

- For **guests**: group-level selection applies (`NOT_LOGGED_IN` group).
- For **logged-in customers**: group-level selection applies only if the customer-level rule is set to **Fallback**.

Group Custom Sort Order

If the group-level rule is set to "Custom sort order":

- Pricesystem uses the group-specific priority list stored in `pricesystem_customer_group_sort_position`
- If the group has no custom list configured, it falls back to the global `price_sort_order`

Customer Overrides + Fallback

Pricelist · /pricelist/price-selection/customer-overrides

Customer overrides require:

- `mageb2b/pricesystem-advancedconfig` (MageB2B_PricesystemCoreAdvancedConfig)

System Config vs Fallback

- "System Config": uses the global strategy and does not fall back to group settings.
- "Fallback": uses group settings first (if configured), otherwise system config.

Recommendation:

- Set `pricesystem/advanced/default_priceselect` to **Fallback** so group-level rules apply by default.

Formula Pricing

Pricelist · /pricelist/price-selection/formula-pricing

Formula pricing requires:

- `mageb2b/pricesystem-advancedconfig` (`MageB2B_PricesystemCoreAdvancedConfig`)

To reference Pricelist in formulas, use:

- `pricelist`
- `base_pricelist` (if you use a base pricelist)

Formula inputs and fallback behavior are controlled by:

- `pricesystem/price_select_rule/formula_missing_input_behavior`
- `pricesystem/price_select_rule/formula_fallback_strategy`

Sort Order Strategy

Pricelist · /pricelist/price-selection/sort-order-strategy

When `pricesystem/price_select_rule/priceselect = Sort order`, Pricesystem selects the first available price from your configured priority list.

To prioritize pricelist pricing, include `pricelist` near the top.

If you use a base pricelist as a fallback price type, consider including `base_pricelist` lower in the list (above `orig_price`).

System Configuration (Global)

Pricelist · /pricelist/price-selection/system-configuration

Global selection strategy is configured in Pricesystem core:

- Admin -> Stores -> Configuration -> MageB2B -> Pricesystem

Global Strategy: Price Select Rule

- `pricesystem/price_select_rule/priceselect` (Lowest / Highest / Sort order; Formula requires Advanced Config Add-On)

Pricelist-Specific Settings

Pricelist adds additional settings under the same Pricesystem section:

- Admin -> Stores -> Configuration -> MageB2B -> Pricesystem -> Pricelist settings

Important keys:

- Download link: `pricesystem/pricelist/enable_download`
- CSV delimiter: `pricesystem/pricelist/csv_delimiter`
- CSV encloser: `pricesystem/pricelist/csv_encloser`
- Merge qty tiers across pricelists: `pricesystem/pricelist/merge_pricelist_qtys`
- Deactivate old pricelists: `pricesystem/pricelist/deactivate_old_pricelists`
- Auto assignment:
`pricesystem/pricelist/enable_auto_assign_customers_to_pricelist`
- Strict matching (auto assignment): `pricesystem/pricelist/match_exact_type`
- Base pricelist validation: `pricesystem/pricelist/enable_base_pricelist_validation`

Merge Pricelist Qtys (Behavior)

- If disabled, prices from pricelists with lower priority are ignored.
- If multiple pricelists share the same highest priority, their qty tiers can still be combined (depending on configuration).