



SOFTWARESILOS

Category Specific Prices

PDF Manual

MODULE

categoryprice

LAST UPDATED

2026-03-29

Table of Contents

Category Specific Prices · categoryprice

- [Overview](#)
- [Category Specific Prices](#)
- [Getting Started](#)
 - [Installation & Setup](#)
 - [Creating Category Prices](#)
- [Features](#)
 - [Category Customer Pricing](#)
 - [Category Group Pricing](#)
 - [Priority System](#)
- [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 \(Global\)](#)

Category Specific Prices

Category Specific Prices · /categoryprice

Overview

The **Category Specific Prices Extension** enables B2B merchants to assign custom pricing at the category level, applying prices to all products within a category simultaneously. This powerful system supports both individual customer pricing and customer group pricing (Wholesale, Retail, VIP, etc.), making it ideal for managing bulk pricing strategies across product lines.

Key Features

Category Customer Pricing

Set custom prices at the category level for specific individual customers. One price entry automatically applies to all products within the category, dramatically reducing administrative overhead.

Customer Group Pricing

Assign category-level pricing to entire customer groups rather than individual customers. Define prices for Wholesale, Retail, VIP, or custom groups you create in Magento.

Priority-Based Resolution

Smart conflict handling when both customer-specific and group-based prices exist for the same category. Configure resolution strategy globally or use numeric priority fields for fine-grained control.

Quantity-Based Tiering

Create graduated pricing structures at the category level with multiple quantity breakpoints. Each tier can have different prices and date ranges for complex promotional strategies.

Time-Based Pricing

Schedule category prices with precise date ranges. Set up seasonal pricing, promotional campaigns, or contract renewals that activate automatically on future dates.

Bulk Product Management

The key efficiency advantage - one price entry affects all products in the category. If a category contains 500 products, setting one category price creates pricing for all 500 instantly.

What You Can Do

- Set category prices for individual customers
- Set category prices for customer groups (Wholesale, Retail, VIP)
- Configure priority-based conflict resolution

- Create quantity-based pricing tiers
- Schedule time-based pricing campaigns
- Set different prices per website/region
- Allow customers to export their prices
- Bulk import/export with add-on
- API access with add-on

Use Cases

Wholesale Category Pricing

Set discounted pricing for entire product categories for wholesale customer groups. One category price applies to 200+ products instantly.

VIP Category Discounts

Reward VIP customers with exclusive pricing on premium product categories using high priority values that override standard group pricing.

Seasonal Category Campaigns

Schedule promotional pricing for entire product categories during peak seasons with automatic activation and deactivation.

Multi-Brand Category Management

Manage pricing for different brands as separate categories with brand-specific customer groups and priority-based conflict resolution.

Regional Category Pricing

Set different category prices for different regions using multi-website support. US wholesale sees USD pricing, EU sees EUR pricing.

Contract-Based Category Pricing

Implement annual contracts with category-level pricing agreements. Use quantity tiers to reward volume commitments at the category level.

Technical Details

Extension: MageB2B_PricesystemCategoryprice **Dependencies:** MageB2B_PricesystemCore >= 2.3.0

Database Tables:

- `pricesystem_categoryprice` - Customer category prices
- `pricesystem_categoryprice_customergroup` - Group category prices **Price Code:** categoryprice

Getting Started

1. **Installation** - Install and configure the extension

2. **Creating Prices** - Add your first category-specific prices
3. **Category Customer Pricing** - Individual customer pricing
4. **Category Group Pricing** - Customer group pricing
5. **Priority System** - Configure conflict resolution
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

Category Specific Prices · [/categoryprice/getting-started/installation](#)

Requirements

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

Installation Steps

Install the module via Composer:

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

# Install the module
composer require mageb2b/pricesystem-categoryprice:*

# Enable the module
php bin/magento module:enable MageB2B_PricesystemCategoryprice

# Run setup upgrade
php bin/magento setup:upgrade

# Compile DI
php bin/magento setup:di:compile

# Deploy static content
php bin/magento setup:static-content:deploy

# Clear cache
php bin/magento cache:flush
```

Configuration

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

Available Settings

Price Select Rule

Choose how to resolve conflicts when both customer and group prices exist.

- **Path:** pricesystem/categoryprice/price_select_rule
- **Type:** Select
- **Options:**

- **Select price by priority** - Use numeric priority field to determine winner
- **Group price first** - Always prefer customer group prices
- **Customer price first** - Always prefer individual customer prices
- **Default:** Select price by priority

This is a critical setting that determines pricing behavior when conflicts occur.

Enable Download Link

Allow customers to download their personal category prices as CSV from their account.

- **Path:** `pricesystem/categoryprice/enable_download`
- **Type:** Yes/No
- **Default:** No

When enabled, customers will see a "Download My Prices" link in their account dashboard showing all category prices that apply to them.

CSV Delimiter

Configure the delimiter used in CSV exports.

- **Path:** `pricesystem/categoryprice/csv_delimiter`
- **Type:** Text
- **Default:** `,` (comma)
- **Examples:** `;` (semicolon), `|` (pipe), `\t` (tab)

CSV Encloser

Configure the enclosure character used in CSV exports.

- **Path:** `pricesystem/categoryprice/csv_encloser`
- **Type:** Text
- **Default:** `"` (double quote)

Verification

After installation, verify the module is active:

```
php bin/magento module:status MageB2B_PricesystemCategoryprice
```

Expected output:

```
Module is enabled
```

Admin Access

Navigate to: **Catalog > Category Prices**

You should see two sub-menus:

- **Customer Prices** - Individual customer category pricing
- **Customer Group Prices** - Group-based category pricing

Both provide management grids for creating and editing category prices.

Database Tables

The extension creates two main tables:

pricesystem_categoryprice

Stores individual customer category prices.

Key Fields:

- entity_id → Category ID (foreign key to catalog_category_entity)
- customer_id → Customer ID (foreign key to customer_entity)
- qty → Quantity tier
- value → Price value
- priority → Numeric priority for conflict resolution
- from_date / to_date → Date range
- website_id → Website scope

pricesystem_categoryprice_customergroup

Stores customer group category prices.

Key Fields:

- entity_id → Category ID (foreign key to catalog_category_entity)
- group_id → Customer Group ID (foreign key to customer_group)
- qty → Quantity tier
- value → Price value
- priority → Numeric priority for conflict resolution
- from_date / to_date → Date range
- website_id → Website scope

Next Steps

- [Creating Your First Category Price](#)
- [Category Customer Pricing](#)
- [Category Group Pricing](#)
- [Priority System Configuration](#)

Creating Category Prices

Category Specific Prices · /categoryprice/getting-started/creating-prices

Learn how to create your first category-specific prices for both individual customers and customer groups.

Prerequisites

- Extension installed and configured
- Admin access to **Catalog > Category Prices**
- At least one category in your catalog
- Customers or customer groups created

Two Types of Category Prices

The extension provides two separate interfaces for different pricing strategies:

1. **Customer Prices** - Individual customer category pricing
2. **Customer Group Prices** - Group-based category pricing (Wholesale, Retail, VIP, etc.)

Both types work together using the priority system for conflict resolution.

Creating Customer Category Prices

Step 1: Navigate to Customer Prices Grid

Admin Panel > Catalog > Category Prices > Customer Prices

This shows a grid of all existing customer category prices with columns:

- Category
- Customer
- Quantity
- Price
- Priority
- From Date / To Date
- Website
- Actions (Edit/Delete)

Step 2: Add New Customer Price

Click **Add New** button in the top right.

Step 3: Fill Required Fields

Basic Information

Category:

- Click the category selector
- Navigate the category tree
- Select the target category
- All products in this category will receive this price

Customer:

- Start typing customer name or email
- Select from autocomplete dropdown
- Or enter customer ID directly

Quantity:

- Minimum quantity for this price tier
- Example: 1 for first tier
- Default: 1

Price:

- Custom price value
- Example: 99.99
- This overrides the regular category price

Optional Fields

Priority:

- Numeric value (0-999)
- Higher values take precedence
- Default: 0
- Examples:
 - 10 - Standard pricing
 - 20 - Promotional pricing
 - 30 - VIP pricing

From Date:

- Start date for price validity
- Leave empty for immediate activation
- Format: YYYY-MM-DD or use date picker
- Example: 2025-01-01

To Date:

- End date for price validity
- Leave empty for no expiration
- Must be after From Date
- Example: 2025-12-31

Website:

- Select website scope
- Use "All Websites" for global pricing
- Or select specific website for regional pricing

Price Application Type:

- How the price should be applied
- Options:
 - **Fixed** - Absolute price value
 - **Discount** - Percentage discount
 - **Adjustment** - Add/subtract from base price

Step 4: Save

Click **Save** button. The price is now active for the customer and category.

Step 5: Verify

1. Log in as the customer (or use dev headers)
2. Navigate to a product in the category
3. Verify the custom price displays

Creating Customer Group Category Prices

Step 1: Navigate to Customer Group Prices Grid

Admin Panel > Catalog > Category Prices > Customer Group Prices

This shows a grid similar to customer prices but with customer groups instead.

Step 2: Add New Group Price

Click **Add New** button.

Step 3: Fill Required Fields

Basic Information

Category:

- Same as customer prices
- Select category from tree

Customer Group:

- Select from dropdown
- Options include:
 - General
 - Wholesale
 - Retail
 - NOT LOGGED IN
 - Custom groups you created

Quantity:

- Minimum quantity for tier
- Same behavior as customer prices

Price:

- Custom price for entire group
- Example: 89.99 for wholesale group

Optional Fields**Priority:**

- Same as customer prices
- Use higher priority for group prices that should override customer prices
- Example: Set group priority=25 to override customer priority=20

From Date / To Date:

- Same behavior as customer prices
- Schedule group pricing campaigns

Website:

- Same as customer prices
- Regional group pricing

Price Application Type:

- Same options as customer prices

Step 4: Save and Verify

Save and test with a customer in that group.

Creating Quantity Tiers

To create graduated pricing (bulk discounts), add multiple price entries with different quantities:

Example: Wholesale Customer Group Tiers**1. Tier 1 - Small orders**

- Category: Electronics
- Group: Wholesale
- Qty: 1
- Price: 100.00
- Priority: 10

2. Tier 2 - Medium orders

- Category: Electronics
- Group: Wholesale
- Qty: 10
- Price: 95.00
- Priority: 10

3. **Tier 3** - Large orders

- Category: Electronics
- Group: Wholesale
- Qty: 50
- Price: 90.00
- Priority: 10

4. **Tier 4** - Bulk orders

- Category: Electronics
- Group: Wholesale
- Qty: 100
- Price: 85.00
- Priority: 10

When a wholesale customer orders 75 items from Electronics, they get the \$90 price (Tier 3).

Creating Time-Based Pricing

Schedule prices for specific date ranges:

Example: Seasonal Campaign

Summer Sale (June-August):

- Category: Outdoor Furniture
- Group: General
- Qty: 1
- Price: 199.99
- Priority: 20 (higher than standard pricing)
- From Date: 2025-06-01
- To Date: 2025-08-31
- Website: All Websites

This price activates automatically on June 1st and expires after August 31st.

Priority System in Action

Scenario: Conflict Resolution

Setup:

1. Customer "John Doe" is in "Wholesale" group
2. Create customer price: John + Electronics + Qty 1 + \$95 + Priority 10
3. Create group price: Wholesale + Electronics + Qty 1 + \$90 + Priority 20

Result:

- Group price wins (priority 20 > 10)
- John pays \$90

Configuration Override

Navigate to: **Stores > Configuration > Pricesystem > Categoryprice Settings > Price Select Rule**

- **Select by priority** - Use priority field (default)
- **Customer price first** - Always prefer customer prices (ignores priority)
- **Group price first** - Always prefer group prices (ignores priority)

Change this setting to control global behavior.

Bulk Product Coverage

Understanding the Efficiency

Traditional Product Pricing:

- Category "Electronics" has 500 products
- Set custom price on 500 products individually
- Total entries: 500

Category Pricing:

- Category "Electronics" has 500 products
- Set custom price on 1 category
- Total entries: 1

New Products Auto-Inherit

When you add a new product to a category that has category pricing:

1. Product is assigned to category
2. Product automatically inherits category price
3. No manual price entry needed

This is the key advantage for growing catalogs.

Common Workflows

Workflow 1: VIP Customer

Goal: Give customer #12345 special pricing on Premium Products category

1. Navigate to **Catalog > Category Prices > Customer Prices**
2. Click **Add New**

3. Category: Premium Products
4. Customer: Search "12345" or name
5. Qty: 1
6. Price: 299.99
7. Priority: 30 (high priority for VIP)
8. Save

Workflow 2: Wholesale Group Discounts

Goal: 15% discount for wholesale customers on all categories

For each major category:

1. Navigate to **Catalog > Category Prices > Customer Group Prices**
2. Click **Add New**
3. Category: [Category Name]
4. Group: Wholesale
5. Qty: 1
6. Price Application Type: Discount
7. Price: 15 (15% off)
8. Priority: 15
9. Save

Repeat for all categories or use Import/Export add-on for bulk.

Workflow 3: Promotional Campaign

Goal: Black Friday sale on all categories for all customers

For each category:

1. Navigate to **Catalog > Category Prices > Customer Group Prices**
2. Click **Add New**
3. Category: [Category Name]
4. Group: General
5. Qty: 1
6. Price: [Discounted price]
7. Priority: 25
8. From Date: 2025-11-29
9. To Date: 2025-12-02
10. Save

Prices activate automatically on Black Friday and expire after Cyber Monday.

Grid Management

Filtering

Use grid filters to find prices:

- Filter by Category

- Filter by Customer/Group
- Filter by Date Range
- Filter by Website
- Filter by Priority

Mass Actions

Select multiple rows and:

- Delete selected prices
- Export selected to CSV (with Import/Export add-on)

Inline Editing

Click directly on cells to quick-edit:

- Priority
 - Price
 - Dates
-

CSV Export (For Customers)

If enabled in configuration, customers can download their prices:

Customer Dashboard > My Prices > Download CSV

CSV contains:

- Category names
- Quantities
- Prices
- Date ranges
- Websites

This helps customers track their custom pricing agreements.

Troubleshooting

Price Not Showing

Check:

1. Date range valid? ($\text{from_date} \leq \text{today} \leq \text{to_date}$)
2. Website matches? (customer on correct website)
3. Customer logged in? (for customer-specific prices)
4. Customer in correct group? (for group prices)
5. Product in correct category?
6. Cache cleared? (Flush Magento cache)

Wrong Price Applied

Check:

1. Priority system configuration
2. Multiple conflicting prices? (check priority values)
3. Product has product-specific price? (might override category price)
4. Pricesystem Core price resolution order

Can't Save Price

Check:

1. All required fields filled?
 2. Date format correct?
 3. Category exists?
 4. Customer/Group exists?
 5. Website ID valid?
 6. Admin permissions?
-

Next Steps

- [Category Customer Pricing Details](#)
- [Category Group Pricing Details](#)
- [Priority System Configuration](#)
- [Import/Export Guide](#)

Category Customer Pricing

Category Specific Prices · /categoryprice/features/category-customer-pricing

Set custom prices at the category level for individual customers. One price entry affects all products in the category.

Overview

Category Customer Pricing allows you to assign specific prices to individual customers for entire product categories. This is the most granular level of category pricing control.

Key Characteristics

- **Individual customer targeting** - One customer, one category
 - **Database table:** pricesystem_categoryprice
 - **Foreign keys:** customer_id → customer_entity, entity_id → catalog_category_entity
 - **Unique constraint:** category + customer + qty + website + dates + application type
 - **Priority system:** Conflicts resolved by priority field or global rule
-

When to Use Customer Pricing

Use Case 1: Contract-Based Pricing

Scenario: Customer signed annual contract for specific pricing on office supplies.

Solution:

- Category: Office Supplies
- Customer: Customer #12345
- Price: \$19.99 (contracted price)
- From Date: 2025-01-01
- To Date: 2025-12-31
- Priority: 30

All office supply products honor the contract price for this customer.

Use Case 2: VIP Customer Rewards

Scenario: Top customer gets exclusive 20% discount on Electronics.

Solution:

- Category: Electronics
- Customer: VIP Customer #7890
- Price Application Type: Discount
- Price: 20 (20% off)
- Priority: 35

The discount applies to all existing and future electronics products.

Use Case 3: Loyalty Program

Scenario: Customer reached Gold tier, gets special pricing on Premium category.

Solution:

- Category: Premium Products
- Customer: Gold Member #4567
- Price: Reduced by 15%
- Priority: 25

Automatically applies when customer browses premium products.

Use Case 4: Make-Good Pricing

Scenario: Customer had service issue, compensated with reduced pricing for 3 months.

Solution:

- Category: All categories (create multiple entries)
- Customer: Affected Customer #3456
- Price Application Type: Discount
- Price: 10 (10% off everything)
- From Date: 2025-03-01
- To Date: 2025-05-31
- Priority: 20

Temporary discount expires automatically.

Database Structure

Table: pricesystem_categoryprice

Schema:

```
CREATE TABLE pricesystem_categoryprice (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  entity_id INT NOT NULL,           -- Category ID  
  customer_id INT NOT NULL,         -- Customer ID  
  qty DECIMAL(12,4) NOT NULL,       -- Quantity tier  
  value DECIMAL(12,4) NOT NULL,     -- Price value  
  priority INT DEFAULT 0,           -- Priority for conflict resolution  
  from_date DATE DEFAULT NULL,      -- Start date (NULL = immediate)  
  to_date DATE DEFAULT NULL,        -- End date (NULL = permanent)  
  website_id INT NOT NULL,          -- Website scope  
  price_application_type VARCHAR(50), -- fixed|discount|adjustment  
  created_at TIMESTAMP,  
  updated_at TIMESTAMP,
```

```

    FOREIGN KEY (entity_id) REFERENCES catalog_category_entity(entity_id) ON
DELETE CASCADE,
    FOREIGN KEY (customer_id) REFERENCES customer_entity(entity_id) ON DELETE
CASCADE,
    FOREIGN KEY (website_id) REFERENCES store_website(website_id) ON DELETE
CASCADE,

    UNIQUE KEY (entity_id, customer_id, qty, website_id, from_date, to_date,
price_application_type)
);

```

Key Points:

- `entity_id` is the category, not the product
- `customer_id` is the individual customer
- `qty` enables quantity tiers (1, 10, 50, 100, etc.)
- `priority` field for conflict resolution
- Date fields can be NULL for permanent pricing
- Unique constraint prevents duplicate entries

Admin Interface

Location

Admin Panel > Catalog > Category Prices > Customer Prices

Grid Columns

- **ID** - Database record ID
- **Category** - Category name with breadcrumb
- **Customer** - Customer name and email
- **Qty** - Minimum quantity for this tier
- **Price** - Custom price value
- **Priority** - Conflict resolution priority (0-999)
- **From Date** - Start date (empty = immediate)
- **To Date** - End date (empty = permanent)
- **Website** - Website scope
- **Created** - Record creation timestamp
- **Actions** - Edit/Delete buttons

Grid Features

Filtering:

- Search by customer name/email
- Filter by category
- Filter by date range
- Filter by website
- Filter by priority

Sorting:

- Sort by any column
- Multi-column sorting
- Ascending/descending

Mass Actions:

- Delete selected
- Export to CSV (with add-on)

Pagination:

- 20/50/100/200 per page
 - Jump to page number
-

Creating Customer Prices

Standard Entry Form

Fields:**Category (Required):**

- Type: Category selector (tree)
- Validation: Must exist, must be active
- Example: Electronics > Laptops

Customer (Required):

- Type: Autocomplete search
- Search by: Name, email, customer ID
- Validation: Must exist, must be active
- Example: "john.doe@example.com"

Quantity (Required):

- Type: Decimal (12,4)
- Default: 1.0000
- Min: 0.0001
- Example: 1 (first tier), 10 (bulk tier)

Price (Required):

- Type: Decimal (12,4)
- Validation: Must be positive
- Example: 99.99

Priority (Optional):

- Type: Integer (0-999)
- Default: 0
- Higher = takes precedence

- Example: 30 for VIP pricing

From Date (Optional):

- Type: Date picker
- Format: YYYY-MM-DD
- Default: NULL (immediate)
- Example: 2025-01-01

To Date (Optional):

- Type: Date picker
- Must be >= From Date
- Default: NULL (permanent)
- Example: 2025-12-31

Website (Required):

- Type: Dropdown
- Options: All websites or specific website
- Default: Default website
- Example: "US Website"

Price Application Type (Required):

- Type: Dropdown
- Options:
 - **fixed** - Absolute price (default)
 - **discount** - Percentage discount
 - **adjustment** - Add/subtract amount
- Example: fixed

Quantity Tiers

Creating Tiers

Create multiple entries for same customer + category with different qty values:

Example: Volume Discount for Customer #12345

Category	Customer	Qty	Price	Priority
Electronics	#12345	1	\$100.00	10
Electronics	#12345	10	\$95.00	10
Electronics	#12345	50	\$90.00	10
Electronics	#12345	100	\$85.00	10

Resolution Logic:

- Customer orders 5 items → \$100 each (Qty 1 tier)
- Customer orders 25 items → \$95 each (Qty 10 tier)

- Customer orders 75 items → \$90 each (Qty 50 tier)
- Customer orders 150 items → \$85 each (Qty 100 tier)

Tier Selection Algorithm

1. Get all prices for customer + category + website
2. Filter by date validity (from_date <= today <= to_date)
3. Filter by qty (price.qty <= cart_qty)
4. Sort by qty DESC
5. Select first match (highest qty <= cart qty)

Best Practices

1. **Use same priority** for all tiers of same customer
 2. **Consistent date ranges** across tiers
 3. **Logical qty progression** (1, 10, 50, 100 - not random)
 4. **Price decreases with qty** (volume discount pattern)
 5. **Test edge cases** (qty exactly at breakpoint)
-

Date-Based Pricing

Permanent Pricing

Configuration:

- From Date: NULL
- To Date: NULL

Meaning: Price always active, never expires.

Use Case: Standard contract pricing.

Open-Ended Start Date

Configuration:

- From Date: 2025-06-01
- To Date: NULL

Meaning: Activates June 1st, never expires.

Use Case: New contract starting future date.

Fixed Date Range

Configuration:

- From Date: 2025-01-01
- To Date: 2025-12-31

Meaning: Active only during 2025.

Use Case: Annual contract renewal.

Past Dates

Configuration:

- From Date: 2024-01-01
- To Date: 2024-12-31

Meaning: Already expired, won't apply.

Use Case: Historical record keeping.

Priority System

Priority Values

Recommended Range: 0-100

Common Values:

- **0** - Default/fallback pricing
- **10** - Standard customer pricing
- **20** - Promotional customer pricing
- **30** - VIP customer pricing
- **40** - Override everything

Conflict Scenarios

Scenario 1: Customer vs Customer (Different Priorities)

Setup:

- Customer #123 + Electronics + Qty 1 + \$100 + Priority 10
- Customer #123 + Electronics + Qty 1 + \$90 + Priority 20

Result: \$90 wins (priority 20 > 10)

Why: Higher priority always wins when both match.

Scenario 2: Customer vs Group Price

Setup:

- Customer #123 + Electronics + Qty 1 + \$95 + Priority 15 (customer price)
- Wholesale Group + Electronics + Qty 1 + \$85 + Priority 25 (group price)
- Customer #123 is in Wholesale group

Result: Depends on configuration:

- **Select by priority:** \$85 (priority 25 > 15)
- **Customer first:** \$95 (ignores priority)
- **Group first:** \$85 (ignores priority)

Multi-Website Support

Website Scope

Each price entry is scoped to one website (or all websites).

Example Setup:

Website	Customer	Category	Price
US Website	#123	Electronics	\$100 USD
EU Website	#123	Electronics	€90 EUR
UK Website	#123	Electronics	£80 GBP

Same customer, same category, different prices per region.

All Websites Option

Set `website_id = 0` (or select "All Websites") to apply globally.

When to use:

- Single-website stores
- Global pricing strategy
- Testing/development

When NOT to use:

- Multi-currency stores
- Regional pricing differences
- Tax-inclusive vs exclusive regions

CSV Export for Customers

Configuration

Enable Download Link:

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

- **Enable Download** = Yes
- **CSV Delimiter** = , (or ; for European) (`pricesystem/categoryprice/csv_delimiter`)
- **CSV Encloser** = " (double quotes) (`pricesystem/categoryprice/csv_encloser`)

Customer Access

Location: Customer Dashboard > My Prices

The download button text depends on your theme and configuration.

CSV Format

```
name,sku,qty,price,price_type
"Widget Pro",SKU-123,1,99.99,categoryprice
"Widget Pro",SKU-123,10,95.00,categoryprice
```

Use Cases

- Customer tracking their agreements
 - Customer planning bulk orders
 - Customer comparing prices across categories
 - Customer verifying contract terms
-

API Integration

Category customer prices can be managed via WebAPI if you install the **SOAP / REST API Add-On**:

- [Categoryprice SOAP / REST API Add-On](#)
- [Pricesystem SOAP / REST API \(full endpoint list\)](#)

Core endpoints use the prefix `/V1/pricesystem/...`. For category customer prices the key endpoints are:

- GET `/V1/pricesystem/categoryprice/search`
 - POST `/V1/pricesystem/categoryprice`
 - PUT `/V1/pricesystem/categoryprice/:id`
 - DELETE `/V1/pricesystem/categoryprice/:id`
-

Import/Export

Bulk import/export is available via the **CSV Import/Export Add-On**:

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

In Magento's importer (System > Data Transfer > Import) the entity type is:

- **Pricesystem Categoryprice**
-

Best Practices

Naming Conventions

Use consistent priority values across your organization:

- 0-9: System defaults
- 10-19: Standard customer pricing
- 20-29: Promotional pricing
- 30-39: VIP pricing
- 40-49: Contract overrides
- 50+: Emergency overrides

Documentation

Document each price entry:

- Why was it created?
- Who approved it?
- When does it expire?
- What's the renewal process?

Keep external documentation linked to customer records.

Regular Audits

Schedule quarterly reviews:

1. Expired prices (to_date < today)
2. Unused prices (no orders)
3. Conflicting prices (same customer+category+qty with different priorities)
4. Orphaned prices (customer deleted, category deleted)

Performance

For large catalogs (10,000+ price entries):

1. Index regularly
2. Use caching
3. Optimize database queries
4. Consider archiving old prices

Troubleshooting

Price Not Applying

Checklist:

1. ✓ Customer logged in?
2. ✓ Customer ID matches?

3. ✓ Category matches?
4. ✓ Product in category?
5. ✓ Quantity meets minimum?
6. ✓ Date range valid?
7. ✓ Website matches?
8. ✓ Priority correctly set?
9. ✓ Cache cleared?

Debug Steps:

1. Check database directly: `SELECT * FROM pricesystem_categoryprice WHERE customer_id = X AND entity_id = Y;`
2. Check Magento logs: `var/log/system.log`, `var/log/exception.log`
3. Enable Pricesystem Core debug mode
4. Test with different customer
5. Test with different category

Wrong Price Applies

Possible Causes:

1. Multiple matching prices (check priority)
2. Group price overriding (check price select rule)
3. Product-level price overriding (check Pricesystem Core order)
4. Tier price on product (check product admin)
5. Special price on product (check product admin)
6. Catalog price rule (check Marketing > Promotions)

Resolution:

1. Review all prices for customer: Check both customer and group prices
2. Check priority configuration: Stores > Config > Pricesystem > Categoryprice
3. Review Pricesystem Core order: Check which price type has precedence
4. Clear cache: `php bin/magento cache:flush`

Can't Delete Price

Possible Causes:

1. Database foreign key constraint
2. Insufficient admin permissions
3. Price currently in use (active order)

Resolution:

1. Check admin ACL permissions
2. Check for active orders referencing this price
3. Archive instead of delete

Advanced Scenarios

Scenario 1: Customer Migration

Goal: Move customer prices from old customer ID to new customer ID.

SQL:

```
UPDATE pricesystem_categoryprice
SET customer_id = 456 -- new ID
WHERE customer_id = 123 -- old ID
AND entity_id = 789; -- specific category
```

Scenario 2: Category Merge

Goal: Category A merged into Category B, move all prices.

SQL:

```
UPDATE pricesystem_categoryprice
SET entity_id = 200 -- Category B
WHERE entity_id = 100 -- Category A
AND customer_id = 123;
```

Scenario 3: Bulk Priority Update

Goal: Increase priority for all VIP customer prices.

SQL:

```
UPDATE pricesystem_categoryprice
SET priority = priority + 10
WHERE customer_id IN (SELECT entity_id FROM customer_entity WHERE group_id
= 4);
-- Assuming group_id 4 is VIP
```

Scenario 4: Expire All Temp Prices

Goal: End all temporary promotional prices.

SQL:

```
UPDATE pricesystem_categoryprice
SET to_date = CURDATE()
WHERE priority = 20 -- promotional priority
AND to_date IS NULL; -- currently no end date
```

Related Documentation

- [Creating Category Prices](#)
- [Category Group Pricing](#)
- [Priority System](#)
- [Import/Export Guide](#)

Category Group Pricing

Category Specific Prices · /categoryprice/features/category-group-pricing

Set custom prices at the category level for entire customer groups. One price entry affects all group members shopping in that category.

Overview

Category Group Pricing allows you to assign specific prices to customer groups (Wholesale, Retail, VIP, etc.) for entire product categories. This scales customer pricing to groups.

Key Characteristics

- **Group-based targeting** - One group, one category, many customers
 - **Database table:** `pricesystem_categoryprice_customergroup`
 - **Foreign keys:** `group_id` → `customer_group`, `entity_id` → `catalog_category_entity`
 - **Unique constraint:** category + group + qty + website + dates + application type
 - **Priority system:** Conflicts resolved with customer prices by priority or global rule
-

When to Use Group Pricing

Use Case 1: Wholesale Pricing

Scenario: All wholesale customers get 30% off Electronics.

Solution:

- Category: Electronics
- Group: Wholesale
- Price Application Type: Discount
- Price: 30 (30% off)
- Priority: 15

Every customer in Wholesale group automatically gets 30% off all electronics.

Use Case 2: Retail vs Wholesale

Scenario: Different pricing tiers for retail vs wholesale.

Solution:

Retail Customers:

- Category: Office Supplies
- Group: Retail
- Price: \$29.99
- Priority: 10

Wholesale Customers:

- Category: Office Supplies
- Group: Wholesale
- Price: \$19.99
- Priority: 10

Each group sees their appropriate price.

Use Case 3: VIP Program

Scenario: VIP members get premium pricing on luxury categories.

Solution:

- Category: Luxury Goods
- Group: VIP
- Price: Reduced by 20%
- Priority: 25

VIP membership badge grants automatic discounts.

Use Case 4: B2B Customer Segments

Scenario: Construction companies get special pricing on building materials.

Solution:

- Create customer group: "Construction"
- Category: Building Materials
- Group: Construction
- Price: Bulk contractor pricing
- Priority: 20

All construction company accounts get contractor rates.

Database Structure

Table: `pricesystem_categoryprice_customer`group

Schema:

```
CREATE TABLE pricesystem_categoryprice_customer (
  id INT PRIMARY KEY AUTO_INCREMENT,
  entity_id INT NOT NULL,           -- Category ID
  group_id INT NOT NULL,           -- Customer Group ID
  qty DECIMAL(12,4) NOT NULL,      -- Quantity tier
  value DECIMAL(12,4) NOT NULL,    -- Price value
  priority INT DEFAULT 0,          -- Priority for conflict resolution
  from_date DATE DEFAULT NULL,     -- Start date
  to_date DATE DEFAULT NULL,       -- End date
  website_id INT NOT NULL,         -- Website scope
  price_application_type VARCHAR(50), -- fixed|discount|adjustment
```

```

        created_at TIMESTAMP,
        updated_at TIMESTAMP,

        FOREIGN KEY (entity_id) REFERENCES catalog_category_entity(entity_id) ON
DELETE CASCADE,
        FOREIGN KEY (group_id) REFERENCES customer_group(customer_group_id) ON
DELETE CASCADE,
        FOREIGN KEY (website_id) REFERENCES store_website(website_id) ON DELETE
CASCADE,

        UNIQUE KEY (entity_id, group_id, qty, website_id, from_date, to_date,
price_application_type)
);

```

Key Differences from Customer Table:

- `group_id` instead of `customer_id`
- Foreign key to `customer_group` table
- Same other fields (qty, priority, dates, etc.)

Customer Groups in Magento

Default Groups

Magento ships with these groups:

Group ID	Group Name	Default Tax Class
0	NOT LOGGED IN	Retail Customer
1	General	Retail Customer
2	Wholesale	Wholesale
3	Retailer	Retailer

Custom Groups

You can create custom groups:

Admin Panel > Customers > Customer Groups > Add New Customer Group

Examples:

- VIP Customers
- Construction
- Healthcare
- Education
- Government
- Non-Profit

Each group can have:

- Custom name
 - Tax class
 - Category prices (via this extension)
-

Admin Interface

Location

Admin Panel > Catalog > Category Prices > Customer Group Prices

Grid Columns

- **ID** - Database record ID
- **Category** - Category name with breadcrumb
- **Customer Group** - Group name
- **Qty** - Minimum quantity
- **Price** - Custom price value
- **Priority** - Resolution priority
- **From Date** - Start date
- **To Date** - End date
- **Website** - Website scope
- **Created** - Creation timestamp
- **Actions** - Edit/Delete

Grid Features

Same as customer prices:

- Filtering
 - Sorting
 - Mass actions
 - Pagination
 - Inline editing
-

Creating Group Prices

Standard Entry Form

Category (Required):

- Select from category tree
- Example: Electronics > Laptops

Customer Group (Required):

- Dropdown of all groups
- Options: General, Wholesale, Retail, VIP, etc.

- Example: "Wholesale"

Quantity (Required):

- Decimal (12,4)
- Default: 1
- Example: 10 for bulk tier

Price (Required):

- Decimal (12,4)
- Example: 89.99

Priority (Optional):

- Integer (0-999)
- Default: 0
- Example: 15

From Date (Optional):

- Date picker
- NULL = immediate
- Example: 2025-06-01

To Date (Optional):

- Date picker
- NULL = permanent
- Example: 2025-08-31

Website (Required):

- Dropdown
- Example: "US Website"

Price Application Type (Required):

- fixed / discount / adjustment
- Example: discount

Group-Based Quantity Tiers

Creating Wholesale Tiers

Goal: Graduated wholesale pricing for Electronics.

Group	Category	Qty	Price	Priority
Wholesale	Electronics	1	\$90.00	15
Wholesale	Electronics	25	\$85.00	15
Wholesale	Electronics	100	\$80.00	15

Group	Category	Qty	Price	Priority
Wholesale	Electronics	500	\$75.00	15

Result:

- Wholesale customer orders 50 items → \$85 each (Qty 25 tier)
- Wholesale customer orders 200 items → \$80 each (Qty 100 tier)

Creating Retail Tiers

Goal: Small volume discounts for retail customers.

Group	Category	Qty	Price	Priority
Retail	Office Supplies	1	\$25.00	10
Retail	Office Supplies	5	\$24.00	10
Retail	Office Supplies	10	\$23.00	10

Result:

- Retail customer orders 7 items → \$24 each (Qty 5 tier)

Group Assignment

How Customers Get Group Prices

When a customer logs in:

1. Magento loads customer record
2. Customer record has `group_id` field
3. Extension checks `pricesystem_categoryprice_customer` for matching `group_id`
4. Applies group prices to all products in those categories

Assigning Customers to Groups

Manual Assignment:

Admin Panel > Customers > All Customers > Edit Customer

- Navigate to "Account Information"
- Field: "Customer Group"
- Select from dropdown
- Save

Bulk Assignment:

Select multiple customers in grid:

1. Check customers
2. Actions dropdown: "Assign a Customer Group"

3. Select group
4. Submit

Automatic Assignment:

Use third-party extensions or custom code to:

- Auto-assign based on purchase volume
- Auto-assign based on company domain
- Auto-assign based on custom attributes
- Auto-assign based on registration form data

Changing Groups

When you move a customer to different group:

1. Old group prices no longer apply
2. New group prices apply immediately
3. No price history tracking (current group only)

Example:

Customer moves from Retail → Wholesale:

- Loses Retail category prices
 - Gains Wholesale category prices
 - Takes effect on next page load
-

NOT LOGGED IN Group

Special Case

`group_id = 0` is for **anonymous/guest** visitors.

Use Case: Guest pricing for certain categories.

Example:

- Category: Free Samples
- Group: NOT LOGGED IN
- Price: \$0.00
- Priority: 5

Guests can order free samples without logging in.

Note: Most B2B stores require login, so group 0 pricing is rare.

Priority System with Groups

Group vs Group (Same Customer)

Scenario: Customer in multiple groups?

Answer: Magento doesn't support multiple groups. Each customer has exactly ONE group at a time.

Group vs Customer Prices

Scenario: Customer has both group price and individual customer price.

Setup:

- Customer #123 in Wholesale group
- Group price: Wholesale + Electronics + \$90 + Priority 15
- Customer price: Customer #123 + Electronics + \$85 + Priority 25

Result: Depends on configuration:

- **Select by priority:** \$85 (customer price, priority 25 > 15)
- **Customer first:** \$85 (always customer)
- **Group first:** \$90 (always group)

Configuration: Stores > Config > Pricesystem > Categoryprice > Price Select Rule

Multi-Website Group Pricing

Per-Website Pricing

Scenario: Different pricing per website for same group.

Setup:

Website	Group	Category	Price
US Store	Wholesale	Electronics	\$90 USD
EU Store	Wholesale	Electronics	€85 EUR
UK Store	Wholesale	Electronics	£80 GBP

Wholesale customers see region-appropriate pricing.

Shared Group Pricing

Use `website_id = 0` (All Websites) for global group pricing.

When to use:

- Single website store
- Same pricing across all regions
- Development/testing

Date-Based Group Campaigns

Seasonal Group Promotions

Example: Summer Sale for Retail

- Category: Outdoor Furniture
- Group: Retail
- Price Application Type: Discount
- Price: 25 (25% off)
- From Date: 2025-06-01
- To Date: 2025-08-31
- Priority: 20

All retail customers get summer discount automatically.

Flash Sales for VIP

Example: Weekend VIP Flash Sale

- Category: New Arrivals
- Group: VIP
- Price: Early bird pricing
- From Date: 2025-12-01
- To Date: 2025-12-03
- Priority: 30

VIP members get first access at discounted prices.

Contract Renewals

Example: Annual Wholesale Contract

- Category: All major categories
- Group: Wholesale
- Price: Contracted rates
- From Date: 2025-01-01
- To Date: 2025-12-31
- Priority: 15

Auto-renew by creating next year's entries in advance.

CSV Export for Group Analysis

Group Price Reports

Use the **CSV Import/Export Add-On** to export category group prices and review them in Excel/Sheets:

- [Categoryprice CSV Import/Export Add-On](#)
- CLI: `php bin/magento pricesystem:export-categoryprice-customergroup`

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

Cross-Group Comparison

Export prices for all groups, compare in spreadsheet:

Category	Retail Price	Wholesale Price	VIP Price
Electronics	\$100	\$90	\$85
Office	\$25	\$20	\$18

Ensure logical pricing hierarchy.

API Integration

Category **customer group** category prices can be managed via WebAPI if you install the **SOAP / REST API Add-On**:

- [Categoryprice SOAP / REST API Add-On](#)
- [Pricesystem SOAP / REST API \(full endpoint list\)](#)

Core endpoints use the prefix `/V1/pricesystem/...`. For category customer group prices the key endpoints are:

- GET `/V1/pricesystem/categoryprice/customergroup/search`
 - POST `/V1/pricesystem/categoryprice/customergroup`
 - PUT `/V1/pricesystem/categoryprice/customergroup/:id`
 - DELETE `/V1/pricesystem/categoryprice/customergroup/:id`
-

Import/Export for Groups

Bulk import/export is available via the **CSV Import/Export Add-On**:

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

In Magento's importer (System > Data Transfer > Import) the entity type is:

- **Pricesystem Categoryprice Customergroup**
-

Best Practices

Group Naming

Use descriptive names:

- "Wholesale - Construction"
- "VIP - Gold Tier"
- "Education - K-12"
- "Group 1"
- "Test"
- "Misc"

Priority Conventions

Establish group priority hierarchy:

- **10-14:** Retail/General groups
- **15-19:** Wholesale groups
- **20-24:** Special programs
- **25-29:** VIP groups
- **30+:** Emergency overrides

Group Management

Regular audits:

1. Unused groups (no customers assigned)
2. Empty groups (no prices defined)
3. Obsolete groups (old programs)
4. Duplicate groups (consolidate)

Testing New Groups

Before production:

1. Create test group
2. Assign test customer
3. Set test prices on sample categories
4. Log in as test customer
5. Verify pricing displays correctly
6. Test quantity tiers
7. Test date ranges
8. Clear cache and re-test

Advanced Scenarios

Scenario 1: Group Migration

Goal: Merge "Wholesale A" and "Wholesale B" into "Wholesale".

SQL:

```
-- Move all Wholesale A prices to Wholesale
UPDATE pricesystem_categoryprice_customer_group
SET group_id = 2 -- Wholesale
WHERE group_id = 10; -- Wholesale A

-- Move all Wholesale B prices to Wholesale
UPDATE pricesystem_categoryprice_customer_group
SET group_id = 2 -- Wholesale
WHERE group_id = 11; -- Wholesale B

-- Note: May create duplicates, resolve with priority or delete
```

Scenario 2: Bulk Priority Update

Goal: Increase all VIP group priorities by 10.

SQL:

```
UPDATE pricesystem_categoryprice_customer_group
SET priority = priority + 10
WHERE group_id = 4; -- VIP group
```

Scenario 3: Clone Group Pricing

Goal: Copy all Wholesale prices to new "Wholesale Premium" group.

SQL:

```
INSERT INTO pricesystem_categoryprice_customer_group
(entity_id, group_id, qty, value, priority, from_date, to_date,
website_id, price_application_type)
SELECT
entity_id,
5 AS group_id, -- New group ID
qty,
value * 0.95 AS value, -- 5% better pricing
priority + 5 AS priority, -- Higher priority
from_date,
to_date,
website_id,
price_application_type
FROM pricesystem_categoryprice_customer_group
WHERE group_id = 2; -- Wholesale group
```

Scenario 4: Expire All Group Campaigns

Goal: End all promotional group pricing.

SQL:

```
UPDATE pricesystem_categoryprice_customer_group
SET to_date = CURDATE()
WHERE priority BETWEEN 20 AND 24 -- Campaign priority range
AND to_date IS NULL; -- Currently no end date
```

Troubleshooting

Price Not Showing for Group

Checklist:

1. ✓ Customer logged in?
2. ✓ Customer in correct group? (Check customer record)
3. ✓ Group price exists? (Check admin grid)
4. ✓ Category correct?
5. ✓ Quantity meets minimum?
6. ✓ Date range valid?
7. ✓ Website matches?
8. ✓ Cache cleared?

Debug:

```
-- Check customer group
SELECT entity_id, email, group_id
FROM customer_entity
WHERE email = 'customer@example.com';

-- Check group prices for that group
SELECT *
FROM pricesystem_categoryprice_customer_group
WHERE group_id = X
AND entity_id = Y;
```

Different Customers See Different Prices

Expected Behavior!

- Customer in Retail group → sees Retail prices
- Customer in Wholesale group → sees Wholesale prices

Verify:

1. Check both customer group assignments
2. Confirm both group prices exist
3. Test with correct login credentials

Guest Prices Not Working

Issue: Group 0 (NOT LOGGED IN) prices not applying.

Causes:

1. Customer accidentally logged in
2. Session cookie from previous login
3. Cache issue

Resolution:

1. Clear browser cookies
 2. Test in incognito/private mode
 3. Clear Magento cache
 4. Verify group_id = 0 in database
-

Related Documentation

- [Creating Category Prices](#)
- [Category Customer Pricing](#)
- [Priority System](#)
- [Import/Export Guide](#)

Priority System

Category Specific Prices · </categoryprice/features/priority-system>

Smart conflict resolution when multiple category prices match the same customer and category.

Overview

The Priority System determines which price wins when conflicts occur. Conflicts happen when:

- Same customer has multiple prices for same category
- Customer has both individual price AND group price
- Multiple quantity tiers match
- Multiple date ranges overlap

The system resolves these conflicts predictably and transparently.

How Priority Works

Priority Field

Location: Every price record has a `priority` field

Type: Integer (0-999)

Default: 0

Rule: Higher priority wins

Example:

- Priority 10 vs Priority 20 → **Priority 20 wins**
 - Priority 5 vs Priority 50 → **Priority 50 wins**
 - Priority 0 vs Priority 0 → **First match wins** (undefined behavior)
-

Configuration

Global Price Select Rule

Location: Stores > Configuration > Pricesystem > Categoryprice Settings

Setting: "Price Select Rule"

Options:

1. Select price by priority (Default)

- Uses numeric priority field to determine winner

- Higher priority value takes precedence
- Most flexible and powerful option
- Recommended for complex pricing scenarios

2. Group price first

- Always prefers customer group prices over individual customer prices
- Ignores priority field for customer vs group conflicts
- Still uses priority for group vs group or customer vs customer
- Simpler logic, less flexible

3. Customer price first

- Always prefers individual customer prices over group prices
- Ignores priority field for customer vs group conflicts
- Still uses priority for group vs group or customer vs customer
- Good for customer-centric pricing strategies

Conflict Scenarios

Scenario 1: Customer vs Customer (Same Customer)

Setup:

```
Price 1: Customer #123 + Electronics + $100 + Priority 10
Price 2: Customer #123 + Electronics + $90 + Priority 20
```

Resolution:

- **Both match:** Same customer, same category
- **Winner:** Price 2 (\$90) - Priority 20 > 10
- **Configuration:** Doesn't matter (same customer type)

Use Case: Promotional override of standard pricing.

Scenario 2: Group vs Group (Impossible)

Setup:

```
Price 1: Wholesale Group + Electronics + $90 + Priority 15
Price 2: Retail Group + Electronics + $100 + Priority 10
```

Resolution:

- **Conflict:** Never occurs!
- **Why:** Customers belong to exactly ONE group
- **Result:** Only the customer's group price matches

Note: Magento doesn't support customers in multiple groups simultaneously.

Scenario 3: Customer vs Group

Setup:

```
Customer Price: Customer #123 + Electronics + $95 + Priority 15
Group Price: Wholesale + Electronics + $85 + Priority 25
Customer #123 is in Wholesale group
```

Resolution depends on configuration:

With "Select price by priority"

- **Both match:** Customer #123 matches both prices
- **Winner:** Group price (\$85) - Priority 25 > 15
- **Logic:** Priority field determines winner

With "Customer price first"

- **Both match:** Customer #123 matches both prices
- **Winner:** Customer price (\$95)
- **Logic:** Customer always beats group, priority ignored

With "Group price first"

- **Both match:** Customer #123 matches both prices
 - **Winner:** Group price (\$85)
 - **Logic:** Group always beats customer, priority ignored
-

Scenario 4: Quantity Tier Conflicts

Setup:

```
Price 1: Customer #123 + Electronics + Qty 1 + $100 + Priority 10
Price 2: Customer #123 + Electronics + Qty 1 + $95 + Priority 20
```

Cart:

- Customer #123 ordering 5 items from Electronics

Resolution:

- **Both match:** Qty 1 ≤ 5 for both prices
- **Winner:** Price 2 (\$95) - Priority 20 > 10
- **Result:** All 5 items at \$95

Note: Quantity tiers are resolved BEFORE priority. The system first finds all matching qty tiers, then applies priority among them.

Scenario 5: Date Range Conflicts

Setup:

Price 1: Wholesale + Electronics + \$90 + Priority 10
From: 2025-01-01, To: 2025-12-31

Price 2: Wholesale + Electronics + \$85 + Priority 25
From: 2025-06-01, To: 2025-08-31 (Summer sale)

Timeline:

January - May:

- Only Price 1 is active
- Result: \$90

June - August:

- BOTH prices active (dates overlap)
- Conflict!
- Winner: Price 2 (\$85) - Priority 25 > 10
- Result: \$85 (summer sale)

September - December:

- Only Price 1 is active again
- Result: \$90

Key Point: Higher priority summer sale overrides standard pricing during summer months.

Priority Value Recommendations

Standard Priority Hierarchy

0-9: System Defaults

- Priority 0: Fallback pricing
- Priority 5: Base prices

10-19: Standard Pricing

- Priority 10: Regular customer prices
- Priority 12: Regular wholesale prices
- Priority 15: Regular VIP prices

20-29: Promotional Pricing

- Priority 20: Standard promotions

- Priority 22: Seasonal campaigns
- Priority 25: Flash sales

30-39: VIP/Contract Overrides

- Priority 30: VIP contract pricing
- Priority 35: Special agreements

40-49: Emergency Overrides

- Priority 40: Temporary price fixes
- Priority 45: Urgent promotions

50+: Reserved

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

Resolution Algorithm

Step-by-Step Process

When a customer views a product in a category, the system executes:

1. Get all category prices (both customer and group tables)
2. Filter by customer match:
 - Customer-specific prices for this customer ID
 - Group prices for this customer's group ID
3. Filter by category match:
 - entity_id matches category ID of product
4. Filter by website match:
 - website_id matches current website
 - OR website_id = 0 (all websites)
5. Filter by date validity:
 - (from_date IS NULL OR from_date <= today)
 - AND (to_date IS NULL OR to_date >= today)
6. Filter by quantity:
 - price.qty <= cart_quantity
 - Select highest matching qty tier
7. Apply priority resolution:
 - a. If "Select by priority": Sort by priority DESC
 - b. If "Customer first": Customer prices, then groups
 - c. If "Group first": Group prices, then customers

- 8. Select top result:
 - First price after sorting
 - Apply to product

Configuration Examples

Example 1: Customer-Centric Strategy

Business Rule: "Individual customer agreements always take precedence."

Configuration:

- Price Select Rule: "Customer price first"

Behavior:

- Customer price: \$95, Priority 10
- Group price: \$85, Priority 30
- **Winner:** Customer price (\$95)
- **Reason:** Customer always wins, priority ignored

Best For:

- Contract-based businesses
 - Custom negotiated pricing
 - Account management focus
-

Example 2: Volume-Focused Strategy

Business Rule: "Wholesale groups get best prices, individuals get standard pricing."

Configuration:

- Price Select Rule: "Group price first"

Behavior:

- Customer price: \$100, Priority 30
- Group price: \$85, Priority 10
- **Winner:** Group price (\$85)
- **Reason:** Group always wins, priority ignored

Best For:

- Volume discount programs
- Wholesale-first businesses
- Group pricing incentives

Example 3: Flexible Strategy (Recommended)

Business Rule: "Use priority to handle specific cases dynamically."

Configuration:

- Price Select Rule: "Select price by priority"

Behavior:

```
Standard customer: $95, Priority 10
Wholesale group: $90, Priority 15
VIP customer: $85, Priority 30
Summer sale group: $80, Priority 25
```

Results:

- VIP customer wins: \$85 (Priority 30)
- During summer: VIP still wins (30 > 25)
- Non-VIP in summer: Summer sale wins (25 > 15)

Best For:

- Complex pricing strategies
 - Multiple overlapping campaigns
 - Fine-grained control
-

Priority Management

Setting Priorities

When Creating Prices:

1. Consider the price purpose
2. Check existing priorities for similar prices
3. Set new priority accordingly
4. Document priority reasoning

Example Workflow:

1. Check: "What wholesale prices exist?"
2. Find: Priority 12 for standard wholesale
3. New seasonal wholesale: Priority 22 (promotional tier)
4. Document: "Summer wholesale sale, overrides standard wholesale"

Auditing Priorities

Quarterly Review:

```
-- Find conflicting priorities (same customer+category+qty)
SELECT entity_id, customer_id, qty, COUNT(*) as conflicts
FROM pricesystem_categoryprice
GROUP BY entity_id, customer_id, qty
HAVING COUNT(*) > 1;

-- Find conflicting group priorities
SELECT entity_id, group_id, qty, COUNT(*) as conflicts
FROM pricesystem_categoryprice_customergroup
GROUP BY entity_id, group_id, qty
HAVING COUNT(*) > 1;
```

Resolution:

1. Review business reason for conflicts
2. Adjust priorities to match intent
3. Document exceptions

Bulk Priority Updates

Scenario: Increase all VIP priorities by 10.

```
-- Customer prices
UPDATE pricesystem_categoryprice
SET priority = priority + 10
WHERE customer_id IN (
    SELECT entity_id FROM customer_entity WHERE group_id = 4
);

-- Group prices
UPDATE pricesystem_categoryprice_customergroup
SET priority = priority + 10
WHERE group_id = 4;
```

Advanced Scenarios

Scenario A: Three-Tier Override

Setup:

```
Base:      Wholesale + All Categories + $100 + Priority 10
Campaign:  Wholesale + Electronics + $90 + Priority 20
VIP:       Customer #123 + Electronics + $85 + Priority 30
```

Customer #123 in Wholesale group browsing Electronics:

Evaluation:

1. Base wholesale: Matches (\$100, P10)
2. Electronics campaign: Matches (\$90, P20)
3. VIP customer: Matches (\$85, P30)

Winner: VIP price (\$85, Priority 30)

Logic: All three match, highest priority wins.

Scenario B: Date-Based Priority Shift

Setup:

```
Standard: Wholesale + Electronics + $100 + Priority 15
          From: NULL, To: NULL

Black Friday: Wholesale + Electronics + $75 + Priority 30
              From: 2025-11-29, To: 2025-12-02

Cyber Monday: Wholesale + Electronics + $80 + Priority 35
              From: 2025-12-02, To: 2025-12-03
```

Timeline:

November 28:

- Only Standard active: \$100

November 29-30 (Black Friday):

- Standard + Black Friday active
- Winner: Black Friday (\$75, P30 > P15)

December 2 (Cyber Monday):

- Standard + Black Friday + Cyber Monday active
- Winner: Cyber Monday (\$80, P35 > P30)

December 3+:

- Only Standard active: \$100

Key: Temporary campaigns override standard with higher priority, expire automatically.

Scenario C: Customer Override of Group Campaign

Business Rule: "VIP customers get better pricing than group campaigns."

Setup:

Group Campaign: Wholesale + Electronics + \$85 + Priority 25

From: 2025-06-01, To: 2025-08-31

VIP Standard: Customer #123 + Electronics + \$80 + Priority 30

From: NULL, To: NULL

Result:

- VIP customer: Always \$80 (Priority 30 > 25), even during campaign
- Regular wholesale: \$85 during summer campaign

Purpose: VIP customer agreements always honored, even during promotions.

Testing Priority Logic

Manual Testing

Test Case 1: Basic Priority

1. Create: Customer #TEST + Category TEST + \$100 + Priority 10
2. Create: Customer #TEST + Category TEST + \$90 + Priority 20
3. Log in as customer TEST
4. Browse category TEST product
5. **Expected:** \$90 displays
6. **Verify:** Priority 20 won

Test Case 2: Configuration Switch

1. Config: "Select price by priority"
2. Create: Customer #TEST + Electronics + \$95 + Priority 15
3. Create: Wholesale Group + Electronics + \$85 + Priority 25
4. Assign customer TEST to Wholesale
5. **Expected:** \$85 (priority 25)
6. Change config: "Customer price first"
7. **Expected:** \$95 (customer wins, priority ignored)
8. Change config: "Group price first"
9. **Expected:** \$85 (group wins, priority ignored)

Automated Testing

PHPUnit Test Example:

```
public function testPriorityResolution()
{
    $customer = $this->createCustomer();
    $category = $this->createCategory();

    // Create two prices
    $this->createCategoryPrice([
```

```

        'customer_id' => $customer->getId(),
        'entity_id' => $category->getId(),
        'value' => 100,
        'priority' => 10
    );

    $this->createCategoryPrice([
        'customer_id' => $customer->getId(),
        'entity_id' => $category->getId(),
        'value' => 90,
        'priority' => 20
    ]);

    $price = $this->priceResolver->resolve($customer, $category);

    $this->assertEquals(90, $price);
}

```

Troubleshooting

Wrong Price Wins

Symptom: Lower priority price is being applied.

Checklist:

1. ✓ Check configuration: "Select by priority" enabled?
2. ✓ Verify priority values in database
3. ✓ Clear cache: `php bin/magento cache:flush`
4. ✓ Check Pricesystem Core price order
5. ✓ Verify customer/group assignment
6. ✓ Check date ranges (might filter out high priority)

Debug SQL:

```

SELECT
    'customer' as type,
    customer_id,
    NULL as group_id,
    value,
    priority,
    from_date,
    to_date
FROM pricesystem_categoryprice
WHERE customer_id = 123 AND entity_id = 456

UNION ALL

SELECT

```

```
'group' as type,  
NULL as customer_id,  
group_id,  
value,  
priority,  
from_date,  
to_date  
FROM pricesystem_categoryprice_customer_group  
WHERE group_id = 2 AND entity_id = 456  
  
ORDER BY priority DESC;
```

Configuration Not Applying

Symptom: Changed config, no effect.

Steps:

1. Clear config cache: `php bin/magento cache:clean config`
2. Clear full cache: `php bin/magento cache:flush`
3. Verify config saved: Check database `core_config_data` table
4. Check config scope: Store view vs website vs global
5. Re-index if needed: `php bin/magento indexer:reindex`

Undefined Behavior

Symptom: Inconsistent pricing with same priorities.

Cause: Multiple prices with identical priority.

Example:

```
Price 1: $100, Priority 10  
Price 2: $90, Priority 10
```

Problem: Both priority 10, winner undefined (might be insertion order, ID order, or random).

Solution: Assign unique priorities:

```
Price 1: $100, Priority 10  
Price 2: $90, Priority 11
```

Best Practices

1. Use Priority Tiers

Don't use arbitrary values. Use consistent tiers:

- 0-9, 10-19, 20-29, 30-39, etc.

2. Document Priority Meanings

Create internal documentation:

- "Priority 15 = Standard wholesale"
- "Priority 25 = Promotional campaigns"
- "Priority 30 = VIP contracts"

3. Leave Gaps

Don't use consecutive values:

- 10, 15, 20, 25, 30
- 10, 11, 12, 13, 14

Gaps allow inserting priorities later without renumbering.

4. Test Configuration Changes

Before changing global config:

1. Test in staging environment
2. Verify expected behavior
3. Document expected changes
4. Schedule during low traffic
5. Monitor after deployment

5. Audit Regularly

Quarterly review:

- Find conflicts (same priority)
- Find obsolete prices (expired dates)
- Find unused priorities (gaps in sequence)
- Adjust as needed

Related Documentation

- [Creating Category Prices](#)
- [Category Customer Pricing](#)
- [Category Group Pricing](#)

Advanced Config

Category Specific Prices · /categoryprice/addons/advanced-config

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

Package

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

Related

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

CSV Import/Export

Category Specific Prices · [/categoryprice/addons/import-export](#)

Bulk import and export **category prices** via CSV (customer-based and group-based category prices).

Package

- Composer package: `mageb2b/pricesystem-categoryprice-importexport`
- Magento module: `MageB2B_PricesystemCategorypriceImportExport`

Installation

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

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

Admin Import/Export (Magento ImportExport)

Use Magento's importer:

1. Go to **System > Data Transfer > Import**
2. Entity type:
 - **Pricesystem Categoryprice** (customer-specific category prices)
 - **Pricesystem Categoryprice Customergroup** (group category prices)

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

CSV Columns (High Level)

Customer category prices typically include:

- `customer_id` (required): customer ID (in some setups also customer email)
- `category_id` (required)
- `price` (required)
- `qty`, `from_date`, `to_date`, `website_id` (optional)
- `price_application_type` (optional): `fixed`, `surcharge_nominal`, `surcharge_percent`, `discount_nominal`, `discount_percent`
- `price_attribute_id` (optional)

Group category prices typically include:

- `group_id` (required)
- `category_id` (required)
- `price` (required)
- `qty`, `from_date`, `to_date`, `website_id` (optional)

CLI Commands

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

```
php bin/magento pricesystem:import-categoryprice-customergroup  
/path/to/categoryprice_customergroup.csv  
php bin/magento pricesystem:export-categoryprice-customergroup
```

Related

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

SOAP / REST API

Category Specific Prices · [/categoryprice/addons/rest-api](#)

Adds CRUD WebAPI endpoints for **category prices** (customer and customer group category prices).

Package

- Composer package: `mageb2b/pricesystem-categoryprice-api`
- Magento module: `MageB2B_PricesystemCategorypriceApi`

Installation

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

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

Endpoints

Category prices:

Method	URL
GET	<code>/V1/pricesystem/categoryprice/:id</code>
GET	<code>/V1/pricesystem/categoryprice/search</code>
POST	<code>/V1/pricesystem/categoryprice</code>
PUT	<code>/V1/pricesystem/categoryprice/:id</code>
DELETE	<code>/V1/pricesystem/categoryprice/:id</code>

Customer group category prices:

Method	URL
GET	<code>/V1/pricesystem/categoryprice/customergroup/:id</code>
GET	<code>/V1/pricesystem/categoryprice/customergroup/search</code>
POST	<code>/V1/pricesystem/categoryprice/customergroup</code>
PUT	<code>/V1/pricesystem/categoryprice/customergroup/:id</code>
DELETE	<code>/V1/pricesystem/categoryprice/customergroup/:id</code>

Related

- [Pricesystem Core API](#)

Sample Data

Category Specific Prices · /categoryprice/addons/sample-data

Installs demo entities and example **category price** records.

Packages

- Core sample data: mageb2b/pricesystem-sample-data
(MageB2B_PricesystemCoreSampleData)
- Categoryprice sample data: mageb2b/pricesystem-categoryprice-sample-data
(MageB2B_PricesystemCategorypriceSampleData)

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-categoryprice-sample-data:*

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

Related

- [Pricesystem Sample Data](#)

Price Selection

Category Specific Prices · /categoryprice/price-selection

Categoryprice is a Pricesystem price type (categoryprice). Pricesystem calculates multiple candidate price types and then selects the final purchasable price using a configurable strategy.

This section documents:

- the global selection strategy (Lowest / Highest / Sort order / Formula)
- the Categoryprice-internal conflict rule (customer vs customer group category prices)

Quick Links

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

Categoryprice In The Strategy

- Price code: categoryprice

If you want category prices to win, put `categoryprice` near the top of your sort order list.

Customer Group Overrides

Category Specific Prices · /categoryprice/price-selection/customer-group-overrides

Customer group override strategies require:

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

Where To Configure

- Admin -> Customers -> Customer Groups -> Edit

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

Category Specific Prices · /categoryprice/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

Category Specific Prices · /categoryprice/price-selection/formula-pricing

Formula pricing requires:

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

To reference `Categoryprice` in formulas, use:

- `categoryprice`

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

Category Specific Prices · /categoryprice/price-selection/sort-order-strategy

When the global "Price Select Rule" is set to **Sort order**, Pricesystem selects the first available price from your configured priority list.

To prioritize category pricing, include `categoryprice` near the top.

How The Selection Works

1. Pricesystem builds a map of candidate prices by price code.
2. It iterates the configured `price_sort_order` list from top to bottom.
3. The first price code that exists and has a valid value is selected.
4. If nothing matches, it falls back to `orig_price` (Magento original price).

Skip Zero Price

If `pricesystem/price_calculation/skip_zero_price` = Yes, any price that resolves to `0.00` (or lower) is treated as "not set" and skipped.

System Configuration (Global)

Category Specific Prices · /categoryprice/price-selection/system-configuration

Pricesystem configuration lives in the core module (always installed together with Categoryprice):

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

Global Strategy: Price Select Rule

- Config path: `pricesystem/price_select_rule/priceselect`
- Values:
 - 1 = Lowest
 - 2 = Highest
 - 3 = Sort order
 - 4 = Individual price formula (requires Pricesystem Advanced Config Add-On)

Categoryprice Conflict Rule (Customer vs Customer Group)

Categoryprice can be defined for:

- a specific customer, and/or
- a customer group

If both exist for the same category/qty, Categoryprice needs an internal rule to decide which one to use.

Configure it here:

- Config path: `pricesystem/categoryprice/price_select_rule`

Values:

- Select price by priority
- Group price first
- Customer price first

This setting affects how Categoryprice resolves its own internal conflicts before the global Pricesystem strategy selects between price types.