



SOFTWARESILOS

Product-Customer Matrix Prices

PDF Manual

MODULE

productcustomermatrix

LAST UPDATED

2026-03-29

Installation & Configuration

Product-Customer Matrix Prices · /productcustomermatrix/getting-started/installation

Complete installation guide and configuration reference for Product-Customer Matrix Prices extension.

Requirements

System Requirements

- **Magento:** 2.4.x
- **PHP:** 8.1+
- **MySQL/MariaDB:** 5.7+ / 10.3+
- **Composer:** 2.x

Dependencies

Required:

- `mageb2b/pricesystem-core` - Base pricesystem architecture
- `mageb2b/pricesystem-framework` - Magento core framework

Optional (Recommended):

- `mageb2b/pricesystem-pricelist` - Named pricelists (for integration scenarios)
- `mageb2b/pricesystem-categoryprice` - Category-based pricing

Installation

Step 1: Install via Composer

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

Step 2: Enable Extension

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

Step 3: Run Setup

```
php bin/magento setup:upgrade
php bin/magento setup:di:compile
php bin/magento setup:static-content:deploy -f
```

Step 4: Clear Cache

```
php bin/magento cache:clean
php bin/magento cache:flush
```

Step 5: Verify Installation

Check that tables were created:

```
php bin/magento db:status
```

Expected tables:

- `pricesystem_product_customer_matrix`
- `pricesystem_product_customer_matrix_attribute`
- `pricesystem_product_customer_matrix_customer`

Admin menu should show: **Catalog > Product-Customer Matrices**

Configuration

Configuration Path

Admin Panel > Stores > Configuration > Pricesystem > Product-Customer Matrix

Core Settings

Enable Matrix Pricing

Path: `pricesystem/productcustomermatrix/enabled`

Options:

- **Yes (Default):** Matrix pricing active
- **No:** Matrix pricing disabled (fallback to other pricesystem modules)

Effect:

- When disabled, all matrices ignored regardless of active status
 - Useful for testing or temporary disablement
-

Enable Auto-Assign Customers

Path: `pricesystem/productcustomermatrix/enable_auto_assign_customers`

Options:

- **Yes (Recommended):** Automatic customer assignment based on attributes
- **No:** Manual assignment only

How It Works:

When Enabled:

1. Customer logs in
2. System evaluates all active matrices
3. Matches customer attributes against matrix rules
4. Auto-assigns customer to matching matrices

When Disabled:

- Only manual assignments work (via Customers tab in matrix edit)
- Attribute fields on matrix are ignored

Use Case:

- Enable: Automatic, scalable pricing
- Disable: Full manual control (small customer bases)

Match Exact Type

Path: `pricesystem/productcustomermatrix/match_exact_type`

Options:

- **Loose (No - Default):** Partial string matching, case-insensitive
- **Exact (Yes):** Strict equality, case-sensitive

Examples:

Loose Matching:

```
Matrix Company: "ACME"  
Customer Company: "ACME Corporation" → MATCH (contains "ACME")  
Customer Company: "acme corp" → MATCH (case-insensitive)
```

Exact Matching:

```
Matrix Company: "ACME"  
Customer Company: "ACME Corporation" → NO MATCH (not exact)  
Customer Company: "ACME" → MATCH (exact)  
Customer Company: "acme" → NO MATCH (case-sensitive)
```

Recommendation:

- Use **Loose** for flexible matching (handles typos, variations)
- Use **Exact** for security-sensitive scenarios (tax IDs, legal entity names)

Merge Matrix Quantities

Path: `pricesystem/productcustomermatrix/merge_matrix_qtys`

Options:

- **No (Default):** Use highest priority matrix only
- **Yes:** Merge quantity tiers from all matching matrices

Detailed Explanation:

No (Highest Priority Only)

When customer matches multiple matrices, use ONLY the highest priority matrix.

Example:

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

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

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

Yes (Merge Tiers)

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

Example:

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

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

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

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

Enable Matrix Validation

Path: pricesystem/productcustomermatrix/enable_matrix_validation

Options:

- **Yes (Default):** Strict validation on matrix save
- **No:** Allow potentially conflicting configurations

Validations:

1. **Duplicate priorities:** Warn when same priority exists (same website)
2. **Empty attributes:** Require at least one matching attribute OR manual customer assignment
3. **Date logic:** Ensure from_date <= to_date
4. **Attribute relation:** Validate attributes_relation field (AND/OR)

Recommendation: Keep enabled to prevent configuration errors.

Attribute Relation Default

Path: pricesystem/productcustomermatrix/default_attributes_relation

Options:

- **AND (Default):** All filled attributes must match
- **OR:** Any filled attribute matches

Example:

AND Logic:

```
Matrix:
- Customer Group: Wholesale (2)
- Country: US

Customer:
- Group: Wholesale ✓
- Country: US ✓
Result: MATCH (both match)

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

OR Logic:

```
Matrix:
- Customer Group: Wholesale (2)
- Country: US
```

```
Customer:
- Group: Retail (1)
- Country: US ✓
Result: MATCH (country matches, group doesn't matter)

Customer:
- Group: Wholesale ✓
- Country: DE
Result: MATCH (group matches, country doesn't matter)
```

Use Case:

- **AND:** Precise targeting (Wholesale + US + California)
 - **OR:** Broad targeting (Wholesale OR VIP customers)
-

Advanced Settings

Cache Matrix Assignments

Path: `pricesystem/productcustomermatrix/cache_assignments`

Options:

- **Yes (Default):** Cache matrix-customer assignments per session
- **No:** Re-evaluate on every request

Performance Impact:

- Enabled: Faster, recommended for production
 - Disabled: Slower, useful for testing/debugging
-

Priority Hierarchy

Path: `pricesystem/productcustomermatrix/priority_tiers`

Document your priority hierarchy for team reference.

Recommended Hierarchy:

```
0-9: Base/Fallback
10-19: Standard pricing
20-29: Promotional pricing
30-39: VIP/Contract pricing
40-49: Emergency overrides
50+: System overrides
```

Not a system setting, but document internally.

Database Schema

Table: pricesystem_product_customer_matrix

Main matrix container.

```
CREATE TABLE pricesystem_product_customer_matrix (  
  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,  
  attributes_relation VARCHAR(3) DEFAULT 'AND', -- 'AND' or 'OR'  
  
  created_at TIMESTAMP,  
  updated_at TIMESTAMP,  
  
  FOREIGN KEY (website_id) REFERENCES store_website(website_id),  
  INDEX idx_priority (priority),  
  INDEX idx_active_dates (is_active, from_date, to_date)  
);
```

Table: pricesystem_product_customer_matrix_attribute

Customer attribute matching rules.

```
CREATE TABLE pricesystem_product_customer_matrix_attribute (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  matrix_id INT NOT NULL,  
  attribute_code VARCHAR(50) NOT NULL, -- 'group', 'company', 'tax',  
  'postcode', 'region', 'country'  
  attribute_value VARCHAR(255) NOT NULL,  
  
  FOREIGN KEY (matrix_id) REFERENCES  
  pricesystem_product_customer_matrix(id) ON DELETE CASCADE,  
  INDEX idx_matrix_code (matrix_id, attribute_code)  
);
```

Supported Attributes:

- `group` - Customer group ID
- `company` - Company name
- `tax` - Tax/VAT number
- `postcode` - ZIP/postal code
- `region` - State/province/region
- `country` - Country code (ISO 2-letter)

Table: pricesystem_product_customer_matrix_customer

Manual customer assignments with optional date overrides.

```
CREATE TABLE pricesystem_product_customer_matrix_customer (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  matrix_id INT NOT NULL,  
  customer_id INT NOT NULL,  
  from_date DATE DEFAULT NULL, -- Override matrix from_date  
  to_date DATE DEFAULT NULL, -- Override matrix to_date  
  
  FOREIGN KEY (matrix_id) REFERENCES  
pricesystem_product_customer_matrix(id) ON DELETE CASCADE,  
  FOREIGN KEY (customer_id) REFERENCES customer_entity(entity_id) ON DELETE  
CASCADE,  
  UNIQUE KEY uk_matrix_customer (matrix_id, customer_id),  
  INDEX idx_customer (customer_id)  
);
```

Date Override Logic:

- If customer-specific dates set: Use those dates
- If not set: Use matrix-level dates
- Allows per-customer date customization within same matrix

Product-Price Relationship

Uses shared table from pricesystem core:

```
-- Inherited from pricesystem_pricelist_product  
-- Matrix uses same structure via polymorphic relationship  
CREATE TABLE pricesystem_pricelist_product (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  pricelist_id INT NOT NULL, -- Maps to matrix_id for matrices  
  product_id INT NOT NULL,  
  qty DECIMAL(10,2) DEFAULT 1.00,  
  price DECIMAL(20,4) NOT NULL,  
  from_date DATE DEFAULT NULL,  
  to_date DATE DEFAULT NULL,  
  
  FOREIGN KEY (product_id) REFERENCES catalog_product_entity(entity_id) ON  
DELETE CASCADE,  
  INDEX idx_pricelist_product_qty (pricelist_id, product_id, qty)  
);
```

Post-Installation Checklist

After installation, verify:

1. ✓ **Tables created:** Check database for 3 matrix tables
 2. ✓ **Admin menu:** Catalog > Product-Customer Matrices visible
 3. ✓ **Configuration:** All settings appear in Stores > Configuration
 4. ✓ **Permissions:** Admin users can access matrix grid/edit
 5. ✓ **Cache cleared:** Flush all caches
 6. ✓ **Compilation:** DI compiled without errors
-

Troubleshooting

Extension Not Visible in Admin

Check:

1. Module enabled? `php bin/magento module:status`
2. Cache cleared? `php bin/magento cache:flush`
3. Admin permissions? Check role for matrix resources

Tables Not Created

Check:

1. Setup run? `php bin/magento setup:upgrade`
2. Database errors? Check `var/log/system.log`
3. Permissions? Database user needs CREATE TABLE rights

Configuration Not Saving

Check:

1. Config cache cleared? `php bin/magento cache:clean config`
 2. Database writable? Check `core_config_data` table
 3. Errors in logs? Check `var/log/exception.log`
-

Next Steps

- [Creating Matrices](#) - Step-by-step guide to creating your first matrix
- [Matrix Pricing System](#) - Understanding core matrix architecture