



SOFTWARESILOS

Pricelist

PDF Manual

MODULE

pricelist

LAST UPDATED

2026-03-29

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](#)