Customer Specific Prices

PDF Manual

MODULE

customerprice

LAST UPDATED

2026-03-12

Multi-Website Configuration

Customer Specific Prices · /customerprice/features/multi-website

Overview

Manage different customer prices across multiple websites within your Magento installation. Set region-specific pricing, handle multi-currency scenarios, and maintain separate price lists for different storefronts.

How It Works

The `website_id` field determines which website(s) a price applies to:

- **website_id = 0**: Price applies to all websites (global)
- **website_id = 1, 2, 3, etc.**: Price applies only to that specific website

When resolving prices, the system:

1. Checks for prices with matching website_id
2. Falls back to website_id = 0 if no website-specific price exists
3. Applies standard catalog price if no customer price matches

Website ID Reference

Finding Website IDs

Admin Method:

Navigate to: **Stores > All Stores**

View your website structure:

```
Main Website (ID: 1)
  └─ Main Store
     └─ English Store View
     └─ German Store View

EU Website (ID: 2)
  └─ EU Store
     └─ EU English View
     └─ EU German View

US Website (ID: 3)
  └─ US Store
     └─ US English View
```

Database Query:

```
SELECT website_id, code, name FROM store_website;
```

Example output:

```
| website_id | code   | name         |
|------------|--------|--------------|
| 0          | admin  | Admin        |
| 1          | base   | Main Website |
| 2          | eu     | EU Website   |
| 3          | us     | US Website   |
```

Use Cases

Regional Pricing Strategy

Scenario: Different prices for US and EU customers

```
US Website (website_id=3):
Customer: ACME Corp
Product: Widget ABC
Quantity: 1
Price: $100.00
Website: US Website

EU Website (website_id=2):
Customer: ACME Corp (same customer)
Product: Widget ABC
Quantity: 1
Price: €90.00
Website: EU Website
```

Same customer sees different prices based on which website they're browsing.

Multi-Currency Handling

Example: Product with different currency pricing

```
Main Website (USD):
- Price: $100.00
- Website ID: 1

UK Website (GBP):
- Price: £75.00
- Website ID: 4

EU Website (EUR):
- Price: €85.00
- Website ID: 2
```

Global Price with Regional Overrides

Scenario: Default price everywhere, but cheaper in specific region

```
Global Default:
- Customer: Beta Ltd
- Product: Widget ABC
- Quantity: 1
- Price: $100.00
- Website: All Websites (0)

EU Override:
- Customer: Beta Ltd
- Product: Widget ABC
- Quantity: 1
- Price: €80.00
- Website: EU Website (2)
```

Browsing EU website: €80.00 Browsing any other website: $100.00

Admin Setup

Step 1: Create Website-Specific Price

Navigate to: **Catalog > Customer Prices > Add New**

Step 2: Fill Basic Information

```
Customer: Select customer
Product: Select product
Quantity: 1
Price: 100.00
```

Step 3: Select Website

Website Dropdown Options:

- All Websites (ID: 0)
- Main Website (ID: 1)
- EU Website (ID: 2)
- US Website (ID: 3)

Select the specific website this price should apply to.

Step 4: Save

Click **Save** to create the website-specific price.

Advanced Scenarios

Per-Website Quantity Tiers

Example: Different tier structure by region

```
US Website (website_id=3):
qty=1,   price=$100, website_id=3
qty=50,  price=$95,  website_id=3
qty=100, price=$90,  website_id=3

EU Website (website_id=2):
qty=1,   price=€85,  website_id=2
qty=25,  price=€80,  website_id=2
qty=75,  price=€75,  website_id=2
```

Same customer, same product, but different tier breakpoints and currency per region.

Time-Based Regional Campaigns

Example: Seasonal promotion only in EU

```
EU Q1 Campaign (website_id=2):
- qty=1
- price=€70
- from_date=2025-01-01
- to_date=2025-03-31
- website_id=2

US Standard (website_id=3):
- qty=1
- price=$95
- from_date=NULL
- to_date=NULL
- website_id=3
```

EU customers see promotional price Jan-Mar, US customers see standard price year-round.

Shared Customers Across Websites

Important: If a customer can access multiple websites, they can have different prices on each:

```
Customer ID: 456 (Global B2B Customer)

On US Website:
- Sees $100.00 (website_id=3 price)

On EU Website:
- Sees €85.00 (website_id=2 price)
```

```
On APAC Website:
- Sees $110.00 (website_id=4 price)
```

Database Structure

Schema

```
CREATE TABLE pricesystem_customerprice (
    id INT PRIMARY KEY AUTO_INCREMENT,
    entity_id INT NOT NULL,
    customer_id INT NOT NULL,
    qty DECIMAL(12,4) NOT NULL,
    value DECIMAL(12,4) NOT NULL,
    from_date DATE NULL,
    to_date DATE NULL,
    website_id INT NOT NULL DEFAULT 0,
    CONSTRAINT fk_website FOREIGN KEY (website_id)
        REFERENCES store_website(website_id)
        ON DELETE CASCADE,
    CONSTRAINT UNIQUE (entity_id, customer_id, qty, from_date, to_date,
website_id)
);
```

Unique Constraint

You can have:

- Same customer/product/qty with different website_id
- Same customer/product/website with different qty
- Exact duplicate including website_id (avoid)

Foreign Key Constraint

The foreign key ensures:

- You can only use valid website IDs
- If a website is deleted, associated prices are automatically removed
- Data integrity is maintained

Price Resolution Logic

Resolution Order

When a customer views a product, the system:

1. **Find all customer prices** for entity_id + customer_id
2. **Filter by website**:
    - Include prices where website_id = current website
    - Include prices where website_id = 0 (global)
3. **Filter by date** (if from_date/to_date set)

4. **Filter by quantity** (qty <= cart quantity)
5. **Select best price**:
    - Website-specific price (if exists)
    - Global price (website_id=0) as fallback
    - Highest qty tier that matches

Example Resolution

Setup:

```
Customer: 456
Product: 123
Current Website: 2 (EU)
Cart Quantity: 15

Price Entries:
Entry A: qty=1,  price=$100, website_id=0 (global)
Entry B: qty=10, price=$95,  website_id=0 (global)
Entry C: qty=1,  price=€85,  website_id=2 (EU)
Entry D: qty=10, price=€80,  website_id=2 (EU)
```

Resolution:

1. Filter by website: Keep Entry A, B, C, D (all match website 2 or 0)
2. Filter by quantity: Keep Entry A, C, D (qty 10 ≤ 15)
3. Select best price:
    - Website-specific: Entry D (qty=10, €80, website_id=2)
    - **Entry D wins** (website-specific + highest matching qty)

Result: €80.00 applied

Frontend Display

Product Page

Shows website-specific pricing:

```
Product: Widget ABC
Your Price: €85.00

(Viewing EU Website)
```

Cart

Displays active website price:

```
Widget ABC
Website: EU Website
Quantity: 1
Unit Price: €85.00
```

Customer Account

"Download My Prices" CSV includes website info:

```
Product,SKU,Quantity,Price,Currency,Website
Widget ABC,WGT-ABC,1,85.00,EUR,EU Website
Widget ABC,WGT-ABC,1,100.00,USD,US Website
Widget ABC,WGT-ABC,1,95.00,USD,All Websites
```

Best Practices

Website Configuration

Do:

- Use website_id=0 for default/fallback pricing
- Set website-specific prices only when needed
- Document which websites use which currencies
- Test customer experience on each website
- Consider tax implications per region

Don't:

- Create prices for non-existent website IDs
- Forget to set global fallback prices
- Mix currencies on the same website
- Create unnecessary website-specific entries

Regional Strategy

Centralized Pricing

```
All pricing at website_id=0 (global)
```

Good for: Single-currency stores, simple setups

Hybrid Approach

```
Global defaults + regional overrides
```

Good for: Most multi-website scenarios

Fully Separated

```
Every website has its own price entries
```

Good for: Complex multi-currency, independent regions

Troubleshooting

Wrong Price on Website

Check:

1. Current website ID matches price entry
2. Global fallback (website_id=0) exists
3. Customer is logged in
4. Website filter in admin grid
5. Cache cleared after changes

Debug Query:

```
SELECT * FROM pricesystem_customerprice
WHERE entity_id = 123
  AND customer_id = 456
  AND (website_id = 2 OR website_id = 0)
ORDER BY website_id DESC, qty DESC;
```

Price Not Found on Website

Verify:

1. Price entry exists with correct website_id
2. Foreign key constraint hasn't deleted it
3. Website ID is valid in store_website table
4. Customer has access to that website

Duplicate Entry Error

Error Message:

```
Duplicate entry '123-456-1-NULL-NULL-2' for key 'UNIQUE'
```

Cause: Trying to create duplicate entry for same:

- entity_id + customer_id + qty + dates + website_id

Solution:

- Edit existing entry, or
- Change website_id, or
- Change qty, or
- Change date range

Combining Features

Full Feature Combination Example

```
Multi-Website + Quantity Tiers + Time-Based:
```

```
US Website Q1 Campaign:
qty=1,   price=$90,  from=2025-01-01, to=2025-03-31, website_id=3
qty=50,  price=$85,  from=2025-01-01, to=2025-03-31, website_id=3
qty=100, price=$80,  from=2025-01-01, to=2025-03-31, website_id=3

EU Website Q1 Campaign:
qty=1,   price=€80,  from=2025-01-01, to=2025-03-31, website_id=2
qty=25,  price=€75,  from=2025-01-01, to=2025-03-31, website_id=2
qty=75,  price=€70,  from=2025-01-01, to=2025-03-31, website_id=2

Global Fallback (Year-Round):
qty=1,   price=$100, from=NULL, to=NULL, website_id=0
qty=50,  price=$95,  from=NULL, to=NULL, website_id=0
```

Reporting

Prices by Website Query

```
SELECT
    w.name as website,
    c.email as customer,
    p.sku,
    cp.qty,
    cp.value
FROM pricesystem_customerprice cp
JOIN store_website w ON cp.website_id = w.website_id
JOIN customer_entity c ON cp.customer_id = c.entity_id
JOIN catalog_product_entity p ON cp.entity_id = p.entity_id
WHERE cp.website_id = 2
ORDER BY c.email, p.sku;
```

Global vs Website-Specific Count

```
SELECT
    CASE WHEN website_id = 0 THEN 'Global' ELSE 'Website-Specific' END as
type,
    COUNT(*) as count
FROM pricesystem_customerprice
GROUP BY website_id = 0;
```

Coverage Analysis

```
-- Products with website-specific pricing
SELECT
    p.sku,
    GROUP_CONCAT(DISTINCT w.name) as websites
FROM pricesystem_customerprice cp
JOIN catalog_product_entity p ON cp.entity_id = p.entity_id
JOIN store_website w ON cp.website_id = w.website_id
```

```
WHERE cp.website_id > 0
GROUP BY p.sku;
```

Migration Scenarios

Converting Global to Website-Specific

Step 1: Export existing global prices

```
SELECT * FROM pricesystem_customerprice WHERE website_id = 0;
```

Step 2: Create website-specific copies

```
INSERT INTO pricesystem_customerprice
    (entity_id, customer_id, qty, value, from_date, to_date, website_id)
SELECT
    entity_id, customer_id, qty, value, from_date, to_date, 2 as website_id
FROM pricesystem_customerprice
WHERE website_id = 0;
```

Step 3: Adjust website-specific values as needed

Step 4: Optionally remove global entries

Next Steps

- Quantity-Based Pricing Tiers
- Time-Based Pricing
- Bulk Import with Add-On