Best Practices

SEO Meta Templates for Magento 2

These are the recommended ways to configure the SEO Meta Templates extension based on what we've seen work well. We've included a few real-world examples — simple stores, multi-brand setups, and category-specific configurations — so you can see how others are using it. There's also a list of common mistakes people make and how to avoid them.

Recommended configurations and patterns for SEO Meta Templates.

General Guidelines

Do's

- Start with "Fallback Only" override mode if you already have manually optimized meta tags on some products
- Use optional wrapping `[...]` around variables that might be empty — this keeps your meta tags clean
- Keep meta titles under 60 characters and descriptions under 160 characters
- Use the preview feature to test templates against real products before saving
- Create separate templates for different product types using conditions
- Use meaningful template names so you can identify them in the grid later

Don'ts

- Don't stuff keywords — `{name}, {color}, {brand}, {category}, {sku}, {price}` as a meta title hurts more than it helps
- Don't rely on variables that are empty for most products (e.g., `{color}` on digital products) without optional wrapping
- Don't create overlapping templates with the same priority — the result becomes unpredictable
- Don't forget to set store views when running a multi-store setup with different languages

Common Scenarios

Scenario 1: Simple Single-Store Setup

Use case: One store, one language, consistent meta tags across all products.

Configuration: Template Settings:

- Title Max Length: 60
- Description Max Length: 160
- Override Mode: Always Override

Template — Products:

- Meta Title: `{name}[ - {color}] | {store_name}`
- Meta Description: `Buy {name}[ in {color}] at {store_name}[. {category} collection available now.]`
- Meta Keywords: `{name}, {sku}[, {brand}][, {color}]`
- H1: `{name}`
- Priority: 10

Template — Categories:

- Meta Title: `{name} | {store_name}`
- Meta Description: `Browse our {name} collection at {store_name}[. Find great deals on {parent_category} products.]`
- H1: `{name}`
- Priority: 10

Result: Every product and category gets consistent, attribute-driven meta tags. Empty attributes are gracefully handled by optional wrapping.

Scenario 2: Multi-Brand Store with Fallback

Use case: Store with products from multiple brands, some have brand data, some don't.

Template — Branded products (with condition: brand is not empty):

- Meta Title: `{brand} {name}[ - {color}]`
- Meta Description: `Shop {brand} {name} at {store_name}[. Available in {color}.] Browse our full {brand} range.`
- Priority: 5

Template — Generic fallback (no conditions):

- Meta Title: `{name}[ - {color}] | {store_name}`
- Meta Description: `Buy {name} at {store_name}[. Explore our {category} collection.]`
- Priority: 10

Result: Branded products get brand-prominent meta tags. Unbranded products fall through to the generic template. Priority 5 beats priority 10, so branded always wins when it matches.

Scenario 3: Preserving Manual Meta with Gap Filling

Use case: You have 500 products with carefully written meta titles, but 5000 products with nothing.

Configuration:

- Override Mode: Fallback Only

Template — Products:

- Meta Title: `{name}[ - {color}] | {store_name}`
- Meta Description: `{name}[, {color}][, {brand|manufacturer}]. Shop {category} at {store_name}.`
- Priority: 10

Result: The 500 products with existing meta titles keep their hand-crafted values. The other 5000 get auto-generated meta tags from the template. As you optimize more products manually, those automatically "graduate" out of the template.

Scenario 4: Category-Specific Meta Descriptions

Use case: Different product categories need different meta description patterns.

Template — Electronics (condition: category contains "Electronics"):

- Meta Description: `{name} – {sku}. Technical specs and reviews at {store_name}[. {brand} certified dealer.]`
- Priority: 5

Template — Fashion (condition: category contains "Clothing"):

- Meta Description: `{name}[ in {color}][, by {brand}]. Free shipping on {category} at {store_name}.`
- Priority: 5

Template — Default fallback:

- Meta Description: `Buy {name} at {store_name}[. Great prices on {category}.]`
- Priority: 10

Result: Electronics and fashion products each get context-appropriate descriptions. Everything else falls through to the generic template.

Common Mistakes

Mistake: Missing optional wrapping on unreliable variables

Why it's wrong: `Buy {name} in {color}` becomes "Buy Widget in " when color is empty. **Correct approach:** `Buy {name}[ in {color}]` — the " in " part disappears when color is empty.

Mistake: Setting the same priority for overlapping templates

Why it's wrong: When two templates both match a product with equal priority, the result depends on database ordering which may not be consistent. **Correct approach:** Always use distinct priority values. Lower number =

higher priority. Use 5, 10, 15 increments to leave room for future templates.

Mistake: Forgetting store view assignment in multi-store

Why it's wrong: A template set to "All Store Views" with English text will apply to your Dutch store too. **Correct approach:** Create separate templates per language/store view, or set the appropriate store view for each template.

Mistake: Using "Always Override" with existing hand-crafted meta

Why it's wrong: You lose all your carefully written meta tags. **Correct approach:** Start with "Fallback Only" mode. Switch to "Always Override" only when you're confident your templates produce better results than your manual meta tags.

Need More Help?

Documentation:

- All Help Articles - Complete documentation overview

Support:

- Contact Support - Get help from our team

For a complete overview of features and configuration options, see the SEO Meta Templates extension on magmodules.eu

All articles for SEO Meta Templates

Installation

1	Installation using Composer (recommended)
2	Installation using the Adobe Marketplace
3	Install through FTP and SSH

Configuration

1	Configuration Guide
2	Quick Start Guide

Troubleshooting

1	Troubleshooting

Usage Guide

1	**Best Practices (current)**