Shipping Conditions Documentation

Complete PDF Manual

Guide for Magento administrators, technical project owners, and implementation teams.

LAST UPDATED
2026-10-06

SOURCE
Open complete documentation online

EXTENSION
Magento 2 Extension Shipping Conditions

Table of Contents

Shipping Conditions Documentation · shipping-conditions

Overview
Shipping Conditions
Getting Started
Installation
Create your first rule
Features
Shipping rate actions
Customer conditions
Carrier and method targets
Reference
Database Reference
Troubleshooting
Common Issues
Validation and troubleshooting
FAQ

Shipping Conditions

SoftwareSilo Shipping Conditions changes the shipping rates Magento has already collected. A rule can hide a method, leave it visible with an unavailable message, rename it, or change its price.

Rules can be limited by store view, customer group, date, cart and address data, products, or a specific customer. They are managed under **Stores > Shipping Conditions**.

A rule has four parts

1. **Scope** decides when and where the rule is active.
2. **Conditions** decide which quote or address matches.
3. **Targets** select the carrier methods affected by the rule.
4. **Actions** change each selected rate.

Start with Create your first rule. Use one exact carrier and method while testing. An empty target list applies to every shipping method, so broad rules deserve extra care.

Common uses

- hide an express method for a destination or cart type
- rename a technical carrier label for customers
- add a handling amount to one method
- cap a shipping price at a minimum or maximum
- keep a method visible but explain why it is unavailable

The extension does not create carriers or calculate the original rate. Each carrier must already return a method before a Shipping Conditions rule can change it.

Reference

- Choose carrier and method targets
- Configure actions
- Use customer conditions
- Resolve common issues
- Validate rule configuration
- FAQ

Installation

Install the package from the Magento project directory with the Composer access provided by SoftwareSilo:

```
composer require mageb2b/shipping-conditions
php bin/magento module:enable MageB2B_ExtensionManager
MageB2B_ShippingConditions
php bin/magento setup:upgrade
```

For production mode, follow your normal deployment process for compilation and static content.

Verify the module:

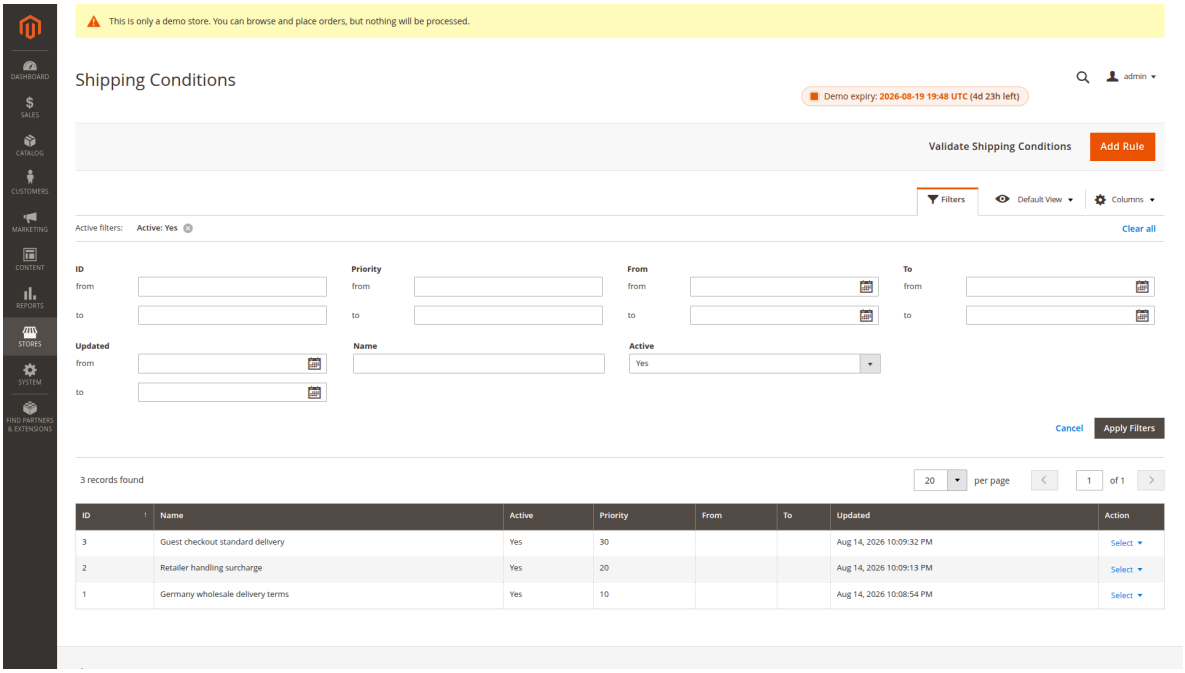
```
php bin/magento module:status MageB2B_ShippingConditions
```

An Admin role with **Shipping Conditions > Manage Rules** permission can now open **Stores > Shipping Conditions**.

The extension has no indexer, cron job or queue consumer to start. Continue with Create your first rule.

Create your first rule

Open **Stores > Shipping Conditions** and select **Add Rule**. Keep the first test narrow so the result is easy to verify.



Example: rename Flat Rate for wholesale customers

1. Enter a clear internal name such as `Wholesale flat-rate label`.
2. Leave **Active** enabled.
3. Select the intended store view and wholesale customer group.
4. Set a unique priority. Lower numbers run first.
5. Leave the date range empty unless the rule is temporary.
6. Leave Conditions empty for this first scope-only rule.
7. Add an exact target for carrier `flatrate` and method `flatrate`.
8. Add **Rename Method Title** with the customer-facing title.
9. Save the rule and test checkout as a customer in that group.

Scope and conditions

Every rule needs at least one store view and customer group. Dates use the selected store's current date and include both endpoints.

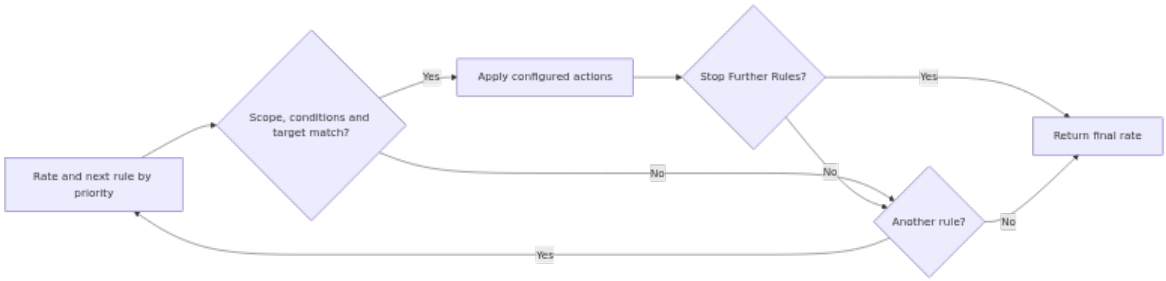
Conditions use Magento's sales-rule builder. An empty condition tree matches every quote address in the selected scope. For a virtual quote, Magento uses the billing address; otherwise it uses the shipping address.

See Customer conditions for the additional customer fields.

Priority and stopping

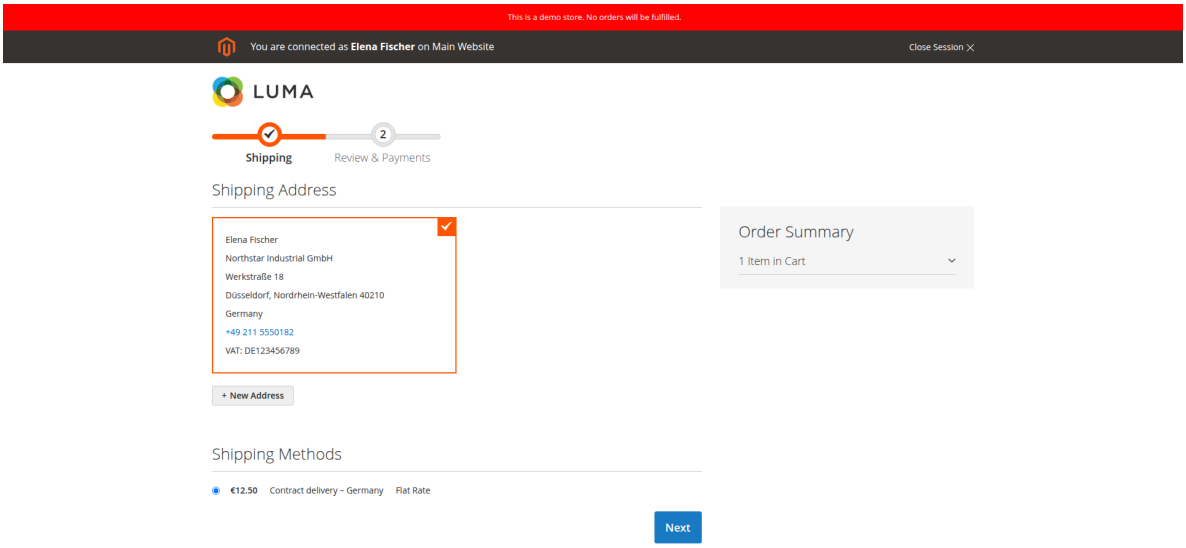
Rules run in ascending priority. Give overlapping rules different priorities because equal values do not provide a reliable tie-breaker.

Stop Further Rules stops later rules for a shipping rate after the current rule has matched its conditions and target. It is evaluated separately for each rate. Use it when a later rule must not alter the same method.



Test before broadening

Check the rule with a matching and non-matching customer, address and cart. Then add more methods, groups or conditions. Run **Validate Shipping Conditions** from the rule grid before go-live.



Shipping rate actions

Actions run in ascending **Sort** order after a rule matches the quote and shipping method. Several price or label actions can be used in one rule.

Action	Effect
Hide Method	Removes the rate from the collected result and ends processing for that rate.
Disable With Message	Stores an error message on the rate so a compatible checkout can show it as unavailable.
Rename Method Title	Changes the method label.
Rename Carrier Title	Changes the carrier label.
Price Fixed (+/-)	Adds the amount to the current price. Use a negative amount to subtract.
Price Percent (+/-)	Adds the percentage of the current price. Use a negative percentage to reduce it.
Replace Price	Replaces the current price with the entered amount.
Cap Min Price	Raises the current price to the amount when it is lower.
Cap Max Price	Lowers the current price to the amount when it is higher.

Action order matters

Each price action sees the result of the preceding action. Adding 10 percent and then replacing the price produces a different result from replacing first and adding 10 percent.

Example for a rate of 20.00:

1. **Price Percent (+/-)** with `10` changes it to 22.00.
2. **Price Fixed (+/-)** with `-2` changes it back to 20.00.
3. **Cap Min Price** with `25` changes the final price to 25.00.

The extension does not impose a zero-price floor. Review negative adjustments so they cannot produce an unintended negative shipping rate.

Disable message presentation

The module transfers the message to the persisted quote address rate. How an unavailable rate is displayed depends on the checkout implementation. Test this action in the actual production theme, including any one-step checkout integration.

Customer conditions

The condition builder includes the standard Magento address and product conditions plus four customer fields:

- Customer ID
- Customer Email
- Customers
- Is Logged In

Customers opens an Admin chooser and stores a list of customer IDs. Use `is one of` to include that list or `is not one of` to exclude it.

Customer ID, email and chooser conditions are most useful for signed-in customers. A guest normally has no customer ID. Use **Is Logged In = No** for a rule intended for guests, then combine it with address or cart conditions if needed.

Example: hide account-only delivery for guests

1. Add **Customer > Is Logged In** and set it to No.
2. Target the account-only carrier and method.
3. Add **Hide Method**.
4. Select all store views and customer groups in which guest checkout can occur.

Test the same cart once as a guest and once as a signed-in customer.

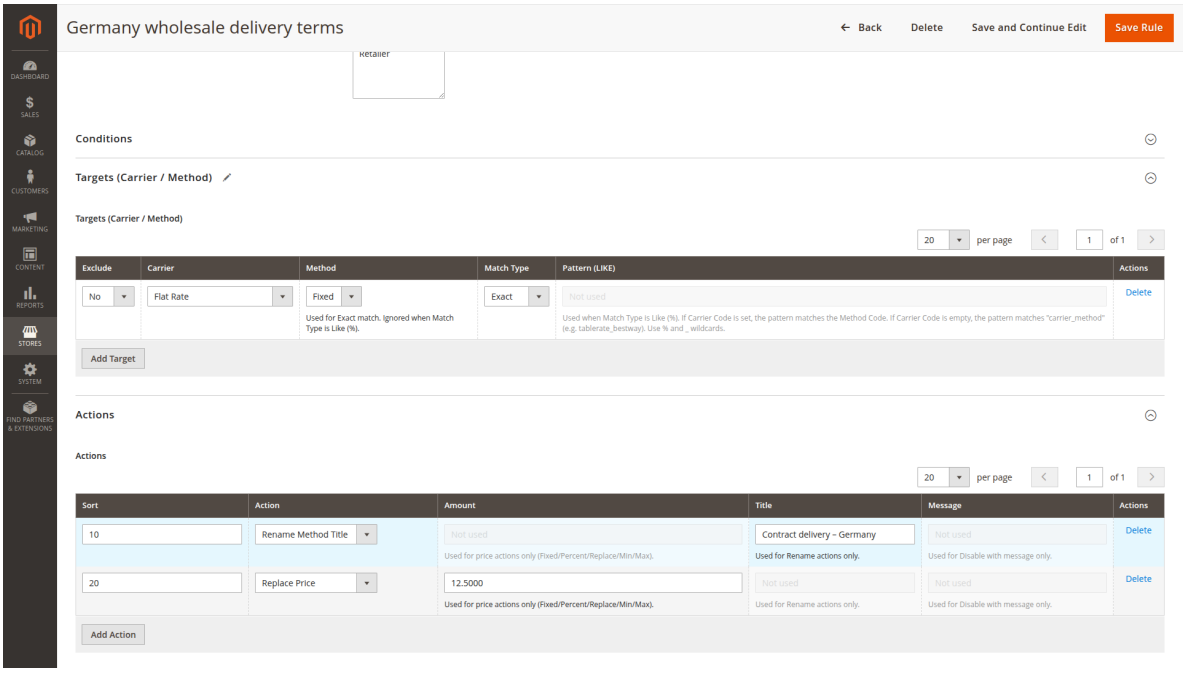
Carrier and method targets

Targets decide which collected shipping rates a rule may change. Exclude rows are checked first, then include rows.

Exact target

Select a carrier and method for the safest match. Common code pairs include `flatrate/flatrate` and `tablerate/bestway`.

If the carrier is empty and a method is entered, that method code can match across carriers. Avoid this broad form unless the shared code is intentional.



Pattern target

Like (%) supports two wildcards:

- `%` matches any number of characters
- `_` matches one character

With a carrier selected, the pattern is compared with the method code. Without a carrier, it is compared with `carrier_method`. Matching is case-insensitive.

Examples:

- carrier `tablerate`, pattern `best%` matches its methods beginning with `best`
- no carrier, pattern `tablerate_%` matches every Table Rate method
- no carrier, pattern `%_freeshipping` matches that method code across carriers

Include and exclude behavior

- An exclude match always removes the rate from the rule's target set.
- If at least one include exists, another rate must match an include to be affected.
- With only excludes, every other rate is affected.
- With no targets, every collected shipping method is affected.

Use exact includes during initial testing. Pattern and exclude-only rules should be checked whenever a carrier integration adds or renames methods.

Database Reference

A Shipping Conditions rule stores its scope, audience, targets and actions separately. This lets one rule address several Store Views, customer groups or shipping methods without duplicating the rule itself.

Relationship	Why it matters
`shipping_condition_rule_store` assigns a rule to one `store`	Controls the Store Views in which the rule can be evaluated.
`shipping_condition_rule_customer_group` assigns a rule to one `customer_group`	Limits the rule to the intended customer audience.
`shipping_condition_rule_target` belongs to one `shipping_condition_rule`	Identifies the shipping methods or rates affected by the rule.
`shipping_condition_rule_action` belongs to one `shipping_condition_rule`	Defines what changes when the rule matches.

Rule conditions and action payloads are deliberately omitted. Manage them through the rule builder so serialization and validation remain intact.

Common Issues

A saved rule does not change checkout

Confirm that the original carrier returned the targeted method. Shipping Conditions changes collected rates; it does not create a missing carrier method.

Then check the rule's active state, date range, store views, customer groups and conditions against the current quote. The carrier and method codes in the target must match the codes returned by Magento, not only the customer-facing labels.

A shipping method disappeared unexpectedly

Review every matching rule in priority order. Look for a **Hide** action, an empty target list that applies broadly, an exclude-only target set or **Stop further rule processing** on an earlier rule. Use one exact carrier and method target while isolating the result.

A rule affects more methods than intended

An empty target list applies to all methods. A target with broad `%` or `_` patterns can also match more codes than its label suggests. Replace it temporarily with an exact carrier and method pair, validate the rule, then widen the pattern only when the exact case works.

A price change is wrong

Check action sort order and all earlier rules that target the same rate. Percentage actions use the rate price at that point in the action chain. A replace action, surcharge and cap can therefore produce different results when reordered.

Magento normalizes localized decimal input when saving. Reopen the rule and confirm the stored amount.

Rename works in Admin but checkout shows the old title

Force the quote to recollect shipping rates by changing or resaving the shipping address, then reload the shipping step. Cached checkout state can retain an older rate result. Also confirm that the checkout theme renders the returned carrier and method titles rather than its own static label.

Disable with message shows no explanation

The action needs a non-empty message. The message is attached to the quote-address rate, but the checkout theme decides how unavailable methods are displayed. Test with the standard checkout or verify the active checkout integration.

Carrier or method options are empty in the rule editor

Select the rule's store views first. The editor loads carrier and method options for that scope. If the selection remains empty, confirm that the carrier is configured for one of those stores, then clean Admin static assets and caches after a module update.

A customer-specific rule does not match

Confirm the signed-in customer ID and website, then check the customer condition separately from the customer-group scope. A changed login state or address can cause Magento to recollect rates with a different quote context.

Two rules override each other

Compare priorities, scopes, conditions and targets. Rules run in priority order, and a matched rule can stop later processing. Give narrow exceptions an earlier priority and broad fallback rules a later one. Run Validate Shipping Conditions to find collisions and overly broad configuration.

Checkout continues after an internal rule error

This is intentional. The collection plugin logs the failure and lets Magento continue with the original rates instead of blocking checkout. Search the Magento logs for `[MageB2B_ShippingConditions] Unable to apply shipping conditions to rates.`

What to collect for support

Provide the rule ID, store view, customer group, destination, cart contents, original carrier and method codes, expected action and final rate. Include the validator output and relevant log entry without credentials or customer secrets.

Validation and troubleshooting

Select **Validate Shipping Conditions** on the rule grid. The validator finds configuration problems such as missing scope assignments, invalid IDs, an inverted date range, incomplete targets or actions, broad rules, and priority collisions.

It reports at most the first 50 warnings. Fix those and run it again. Validation checks stored configuration; it does not simulate carrier responses or checkout rendering.

A rule does not apply

Check in this order:

1. The carrier returned the method before the rule ran.
2. The rule is active and today's store date is within its range.
3. Store view and customer group match the quote.
4. The address, cart, product and customer conditions match.
5. The carrier/method code matches an include and no exclude.
6. An earlier rule did not hide the rate or stop later rules.

A rule affects too many methods

Look for an empty target list, a target with both codes empty, an exclude-only target set, or a broad `%` pattern. Add an exact include while diagnosing.

The final price is unexpected

Review action sort order and every earlier rule that targets the same rate. Percentage actions use the price at that point in the chain, not the carrier's original price in every step.

Localized decimal input is normalized when the rule is saved. Confirm the stored amount after entering a comma decimal separator.

The unavailable message is missing

The message is attached to the quote address rate, but the checkout theme decides how to render unavailable methods. Confirm the rule works with a standard checkout or inspect the active checkout integration.

Checkout still works after a rule error

The rate-collection plugin catches rule exceptions, logs `[MageB2B_ShippingConditions] Unable to apply shipping conditions to rates.`, and leaves rate collection running. Check the Magento log when a rule appears to be ignored without a customer-facing error.

FAQ

Can a rule create a new shipping method?

No. A carrier must return the method first. The rule can then hide, disable, rename or reprice that rate.

What happens when Targets is empty?

The rule applies to every collected shipping method that matches its scope and conditions.

Which rule runs first?

The rule with the lower priority number. Use unique priorities where rules overlap.

Can several rules change the same method?

Yes. Their actions stack in rule order unless a rule hides the method or uses **Stop Further Rules** for that rate.

Can I reduce a shipping price?

Use a negative fixed or percentage amount, or replace the price. Test the result because the module does not add a floor at zero.

Can I target one customer?

Yes. Use Customer ID, Customer Email or the Customers chooser. Keep in mind that guest quotes have no customer ID.

Does the extension expose rules through REST or GraphQL?

No. Rules are maintained in the Magento Admin.

Do I need to reindex after saving a rule?

No. The module evaluates active rules during shipping rate collection. Recollect the cart's shipping rates to test the change.