Flex Shipping Documentation

PDF Manual

MODULE

flex-shipping

LAST UPDATED

2026-03-12

Table of Contents

Flex Shipping Documentation · flex-shipping

- Overview
- Flex Shipping Documentation
- Getting Started
  - Installation
  - Configuration
- Features
  - Methods and Rate Matrix
  - Package Splitting and Scopes
  - Simulator and Audit Trace
- Troubleshooting
  - Common Issues

Flex Shipping Documentation

Flex Shipping Documentation · /flex-shipping

Flex Shipping is a dynamic Magento 2 shipping carrier for method design, package splitting, rule-based calculation, and auditable checkout decisions.

Quick Links

Getting Started

- Installation
- Configuration

Features

- Methods and Rate Matrix
- Package Splitting and Scopes
- Simulator and Audit Trace

Troubleshooting

- Common Issues

Installation

Flex Shipping Documentation · /flex-shipping/getting-started/installation

This guide covers the minimum steps to install and validate Flex Shipping.

1. Install the module

Install via Composer in your Magento instance and run Magento setup commands according to your deployment standard.

2. Enable and verify

After installation:

- Ensure the module is enabled.
- Run setup upgrade and cache refresh.
- Open Magento Admin and confirm Flex Shipping configuration is visible.

3. First smoke check

Create one method and one basic rate, then test checkout once with a sample cart.

4. Recommended next step

Continue with Configuration.

Configuration

Flex Shipping Documentation · /flex-shipping/getting-started/configuration

This page explains the practical setup flow for Flex Shipping.

Step 1: Create shipping methods

For each website, create the required shipping methods and set:

- title
- active status
- sort order
- active time window (if needed)

Step 2: Define attribute mapping

Map weight, volume, and dimension attributes per method. Define how missing values should be handled.

Step 3: Configure package behavior

Choose package splitting strategy and packing mode to match your product and logistics model.

Step 4: Build rate matrix

Define rates with country, postcode, weight, and volume ranges. Use priorities for clear resolution.

Step 5: Set pricing mode and scope

Choose fixed, per kilogram, or per package pricing. Decide whether calculation runs on shipment scope or package scope.

Step 6: Add method rules

Apply Magento conditions and stop-processing logic to keep method behavior deterministic.

Step 7: Validate with simulator

Run simulation for realistic quotes and verify returned methods and calculated amounts before go-live.

Methods and Rate Matrix

Flex Shipping Documentation · /flex-shipping/features/methods-and-rates

Flex Shipping separates method design from pricing logic so operations can adapt without code changes.

What you configure per method

- website scope
- method identity (code/title)
- active status and active windows
- sorting and operational priority

What you configure per rate

- country and postcode range
- weight and volume range
- rule priority
- active period

Why this matters

This setup supports tariff models that differ by region and cart structure while remaining manageable for support and operations teams.

Package Splitting and Scopes

Flex Shipping Documentation · /flex-shipping/features/package-splitting

Flex Shipping supports two calculation models and a configurable package-building logic. This is the key to predictable pricing when carts contain mixed, bulky, or quantity-heavy products.

Calculation scopes

Shipment scope

The full cart is evaluated as one shipment. A matching shipment-level rate is selected and priced once.

Package scope

The cart is split into packages first. Then each package is matched against package-scope rates and priced individually. The final shipping amount is the sum of all package prices.

How package building works

Package creation is driven by method configuration and product-level packing attributes.

Supported pack modes

- `normal`: items are packed together according to your split strategy and package limits
- `always_single`: each unit becomes its own package
- `separate_max_n`: units are split into packages with a configurable max quantity per package

Supported split strategies

- `First Fit Decreasing (ffd)`
- `weight_first`
- `volume_first`
- `custom` (with configurable custom sort order)

Package limits

You can enforce:

- max package weight
- max package volume

If limits are set, package building respects both constraints. If no limits are set, normal-mode items can stay in a single aggregated package.

Missing attribute handling

Package logic depends on product data (weight, dimensions/volume, packing mode, optional separate-max quantity).
If values are missing, behavior is controlled by the method policy:

- strict/block behavior: calculation is stopped
- default-value behavior: configured fallback values are used

This allows either strict data quality enforcement or controlled fallback operation.

Important edge behavior

- Parent/child quote item handling avoids double counting.
- Fractional quantities are normalized for package math.
- If a product uses special pack modes with fractional quantity, logic falls back to normal packing to prevent invalid package states.
- If no valid package can be built, the method does not return a shipping price.

Pricing impact

Package scope changes the commercial outcome:

- Shipment scope: one rate decision for the full cart.
- Package scope: multiple rate decisions, one per package, then sum.

This is especially useful when one cart contains products that should not share the same transport cost model.

Recommended setup sequence

1. Set attribute mapping for weight/volume/packing mode.
2. Define split strategy and package limits.
3. Choose calculation scope per method (shipment or package).
4. Create matching rates for the selected scope.
5. Validate realistic carts with the simulator before go-live.

Typical scenarios

- mixed carts with lightweight and bulky products
- products that must ship separately
- B2B orders where package composition drives transport pricing

Operational result

You get a shipping result that reflects real package structure, not only cart totals. This improves pricing fairness, reduces manual shipping corrections, and increases checkout consistency.

Simulator and Audit Trace

Flex Shipping Documentation · /flex-shipping/features/simulator-and-audit

Flex Shipping includes tools for validation and post-check transparency.

Simulator

Run method calculation against concrete quotes in admin and inspect returned methods and amounts before changing live behavior.

Audit trace

Audit entries keep method-level decision context for each calculation (for example conflict strategy, package metrics, and payload depth) plus the calculated amount. A separate selected `rate_id` is not persisted.

Operational benefit

Support and QA teams can reproduce and explain shipping outcomes faster, which reduces troubleshooting cycles.

Common Issues

Flex Shipping Documentation · /flex-shipping/troubleshooting/common-issues

No Flex Shipping method returned in checkout

Check method active status, website scope, and active window. Then verify rate ranges and method conditions.

Unexpected rate selected

Review conflict strategy and priority settings for overlapping rates.

Shipping amount differs from expectation

Confirm calculation scope (shipment vs package), package splitting strategy, and attribute mappings for weight/volume.

Hard to explain result to support team

Use simulator output and audit trace together to verify and communicate the exact decision path.