magmodules

channable

Shipments

Channable Connect for Magento 2

This guide explains how shipment updates work in the Channable module. It covers what Magento sends back to Channable when you ship an order, how tracking and return labels are recognised, and how the delivery bill ID works — including how to supply your own number from an ERP system.

How Shipment Updates Work

Channable periodically asks your Magento store which orders have been shipped. The module answers with a fulfillment payload per shipment, containing the tracking information and the delivery bill ID. Channable passes this on to the marketplace, which notifies the customer.

Everything is based on standard Magento shipments. As soon as a shipment exists — created manually in the admin, by a shipping extension such as SendCloud, MyParcel or Paazl, or through the REST API — it is picked up automatically. No extra configuration is required.

A shipment update contains:

Field	Source
`tracking_code`	The tracking numbers of the shipment
`carrier_code`	The carrier of each tracking entry
`title`	The title of each tracking entry
`return_tracking_code`	Tracking numbers recognised as a return label
`return_transporter`	The carrier of those return labels
`delivery_bill_id`	The number of the delivery note in the parcel
`shipment_id`	The Magento shipment increment ID

Orders with multiple shipments produce one update per shipment. Each shipment keeps its own tracking data and its own delivery bill ID.

Delivery Bill ID

Some marketplaces — Conrad among them — invoice the customer themselves while you send the parcel. To let the customer link that invoice to the parcel they received, the number of the delivery note inside the parcel is sent along with the shipment. That number is the delivery bill ID.

Default: the Magento shipment increment ID

By default the module sends the Magento shipment increment ID (for example `300000004`). This works without any configuration and requires no manual input:

- It is shown in the admin as soon as a shipment is created
- It is included in the shipment confirmation email
- It is printed on Magento's standard packing slip, so the number on the paper in the parcel matches the number the marketplace receives

For merchants who ship from Magento, this is all you need.

Using your own number from an ERP system

If you print your delivery notes from an ERP or fulfillment system, that system generates its own delivery note number (for example `DSN-123`). In that case the marketplace must receive your number, not Magento's, otherwise the number on the invoice will not match the paper in the box.

Your ERP can supply the number through the `channable_delivery_bill_id` extension attribute on the shipment. There are two ways to do this.

When creating the shipment — add the attribute to the shipment arguments:

```
POST /rest/V1/order/{orderId}/ship

{
  "tracks": [
    {
      "carrier_code": "dhl",
      "title": "DHL",
      "track_number": "3S123456789"
    }
  ],
  "arguments": {
    "extension_attributes": {
      "channable_delivery_bill_id": "DSN-123"
    }
  }
}
```

On an existing shipment — useful when a shipping extension already created the shipment:

```
POST /rest/V1/shipment

{
  "entity": {
```

```
    "entity_id": 12,
    "order_id": 34,
    "extension_attributes": {
      "channable_delivery_bill_id": "DSN-123"
    }
  }
}
```

The value is stored on the shipment and returned again by `GET /rest/V1/shipment/{id}`, so you can always verify what was sent.

When the field is empty, the module falls back to the Magento shipment increment ID. Shipments created by shipping extensions that know nothing about this field therefore keep working exactly as before.

Important: the number you supply must be the same number printed on the delivery note in the parcel. That is the whole purpose of the field.

Return Labels

Shipping extensions often add a second tracking entry for the return label. The module can recognise those entries and report them separately as `return_tracking_code` instead of as a regular tracking code.

Location: Stores → Configuration → Channable → Marketplace → Orders

Return Label Recognition

Choose how return labels are identified. When set to pattern matching, you define per carrier which tracking titles indicate a return label.

Return Label Patterns

A table with two columns:

- **Carrier** — the carrier the rule applies to, or "all" for every carrier
- **Title pattern** — the text that must appear in the tracking title, for example `Return` or `Retour`

A tracking entry matching one of these rules is sent as a return label. All other entries are sent as regular tracking codes.

Multiple Shipments Per Order

When an order is shipped in parts, each shipment is reported separately, with its own tracking data, its own `shipment_id` and its own delivery bill ID. This lets the marketplace tell the customer exactly which items were in

which parcel.

If you use your own delivery note numbers, make sure each partial shipment gets its own number — the same number on two parcels makes the invoice impossible to match.

Checking What Is Sent

The shipment data Channable receives is available through the module's webhook URLs, which you can find under Stores → Configuration → Channable → Marketplace. Opening the shipments URL in your browser shows exactly what Channable sees, including the delivery bill ID per shipment. This is the quickest way to verify your ERP integration is filling the field correctly.

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 Channable Connect extension on magmodules.eu

All articles for Channable Connect

Installation

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

Configuration

1	Quick Start Guide
2	Product Feed Configuration
3	Orders Configuration
4	Product Updates Configuration

Troubleshooting

1	The price of at least one product is missing a currency
2	Test Feed
3	Test Orders

Other

1	**Shipments (current)**

Grids

1	Returns Grid

2	Product Updates Grid
3	Orders Grid

Background

1	How does the Return process work
2	Best Practices
3	CLI Commands

