



Supported entity updates

Workflow

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

LAST UPDATED
2026-08-31

SOURCE
[Open this article online](#)

EXTENSION
[Magento 2 Extension Workflow](#)

Supported entity updates

An Update Entity node changes one supported field on the Magento record identified by the run payload. The builder limits the entity and field choices to the built-in catalogue. This is different from **Set Variable**, which changes only the workflow payload.

Supported records and fields

Entity	Payload key	Selectable fields
Customer	customer_id	Email, customer group, first name, last name, middle name, prefix, suffix, date of birth, gender, Tax/VAT number
Customer Address	address_id	Name fields, company, telephone, fax, postal code, city, VAT number, street, region ID, country, default billing, default shipping
Order	order_id	Status, customer note
Invoice	invoice_id	Comment
Product	product_id	Name, status, visibility, price, special price, weight
Category	category_id	Name, enabled state, menu inclusion, description and search metadata
Inventory Source Item	sku and source_code	Quantity, stock status
CMS Page	page_id	Title, identifier, content, content heading, search metadata, page layout, sort order, enabled state
CMS Block	block_id	Title, identifier, content, enabled state

Product and category changes may be applied at the Store View selected on the node. Other entities use the scope supported by their Magento service contract.

Identifier requirements

The keys in the table belong inside the payload's **entity** object. The node does not search for a record by a display label, email or order increment number. Magento event mappers provide identifiers for supported event types. An API caller must supply them explicitly.

Example order payload:

```
{
  "entity": {
    "entity_type": "sales_order",
    "order_id": 248
  },
  "data": {
    "review_note": "Checked by purchasing"
  }
}
```

The node can then select **Order**, choose **Customer Note**, and render a value from the payload. It cannot update order 248 when only an increment number such as **000000248** is present.

Values and templates

Text values may include payload placeholders supported by the workflow renderer. Select fields use the values offered by the builder. Numeric and date fields still pass through the target Magento service, which may reject an invalid value.

Check the run-node output after a controlled live run. Simulation reports the intended update but suppresses the database write.

Business safeguards

Use the narrowest entity and field that solves the process. Changing a product price, order status, inventory quantity or customer group can affect storefront behavior outside the workflow itself.

Before publishing:

1. confirm that the trigger always supplies the required identifier;
2. test missing and invalid identifiers;
3. test the target field with a disposable Magento record;
4. check any indexing, cache or downstream integration affected by the change;
5. give publish permission only to roles allowed to automate that business action.

An Order status update does not create an invoice, capture a payment, ship an order or issue a refund. Use the Magento operation that owns those actions when the process needs more than a field change.

Related guides

- [Action nodes](#)
- [Triggers and event payloads](#)
- [Validation and simulation](#)

- **Permissions and security**