



## Product Schema

Rich Snippets Suite for Magento 2

Comprehensive guide to configuring Product structured data with the [Rich Snippets Suite](#) extension. Product schema is the most important schema for e-commerce, enabling rich results in Google Search with prices, availability, ratings, and more.

## What Product Schema Does

---

Product schema tells Google about your products:

- **Name and description** - What the product is
- **Images** - Visual representation
- **Price and currency** - Cost information
- **Availability** - In stock or out of stock
- **Condition** - New, refurbished, or used
- **Brand** - Manufacturer or brand name
- **Ratings and reviews** - Customer feedback
- **GTIN/SKU/MPN** - Product identifiers

This information can appear as rich results in Google Search, showing star ratings, prices, and stock status directly in search listings.

## Basic Setup

---

**Location:** Magmodules → Rich Snippets Suite → Schema: Products

1. Set **Enable** to **Yes**
2. Configure **Show Stock** to display availability
3. Set up **Attributes** mapping (Description, Brand)
4. Click **Save Config**

## Offer Schema (Price & Availability)

---

The Offer schema contains purchase-related information.

### Stock Status

Setting	Description
---------	-------------

<b>Show Stock</b>	Enable to include <code>availability</code> in the offer. Google shows "In Stock" or "Out of Stock" in results.
-------------------	---

## Product Condition

Option	When to Use
<b>Fixed value</b>	All products have the same condition (most common for new items)
<b>Use attribute</b>	Different products have different conditions

### Condition Values:

- `NewCondition` - Brand new products
- `RefurbishedCondition` - Factory refurbished
- `UsedCondition` - Second-hand items

## Price Valid Until

Google requires `priceValidUntil` for Offer schema. Options:

Mode	Description
<b>Use special price date</b>	Uses Magento's special price end date if set
<b>Custom fallback</b>	Always use a date X days in the future

**Tip:** Set a custom fallback (e.g., 30-90 days) to ensure all products have valid dates.

## Ratings & Reviews

---

Display star ratings and review counts in search results.

## Aggregate Rating

Setting	Description
<b>Display Rating</b>	Include overall rating in Product schema
<b>Rating Source</b>	Where ratings come from (Native Magento or third-party platform)
<b>Rating Metric</b>	How to display the rating scale

## Individual Reviews

Setting	Description
<b>Add Separate Review(s)</b>	Include individual Review schemas
<b>Number of reviews</b>	How many reviews to include (default: 10)

**Best Practice:** Enable both aggregate rating and individual reviews for maximum visibility.

## Attribute Mapping

---

Map your Magento attributes to schema.org properties.

### Required Mappings

Schema Property	Recommended Attribute
<b>Description</b>	Short Description
<b>Brand</b>	Manufacturer

### Extra Attributes

Add custom mappings for additional product data. The following Schema.org properties are available:

Schema Property	Description
<b>description</b>	Product description (if different from the main description field)
<b>model</b>	Product model name or number
<b>sku</b>	Stock Keeping Unit identifier
<b>gtin8 / gtin12 / gtin13 / gtin14</b>	Global Trade Item Numbers (barcodes). Use <code>gtin13</code> for EAN, <code>gtin12</code> for UPC
<b>mpn</b>	Manufacturer Part Number
<b>isbn</b>	International Standard Book Number (for books)
<b>category</b>	Product category as a Schema.org property. Maps your attribute value directly to the <code>category</code> field
<b>material</b>	Primary material of the product (e.g., "Oak", "Polyester", "Stainless Steel")

Schema Property	Description
<b>color</b>	Product color (e.g., "Anthracite", "Natural Teak")
<b>numberOfItems</b>	Number of items included in the product (e.g., "6" for a dining set with 6 chairs). Useful for sets, bundles, and multi-packs

The `category`, `material`, `color`, and `numberOfItems` properties are available since v2.10.0.

**How `numberOfItems` works:** This Schema.org property tells search engines how many individual items are included in a product. The extension reads the value from your selected Magento attribute, extracts the numeric portion, and outputs it as an integer on the Product schema. This means attribute values like "6 pieces" or "6 stuks" are automatically converted to `"numberOfItems": 6`. This is particularly useful for furniture sets (e.g., a dining set with table + 6 chairs = `"numberOfItems": 7`), multi-packs, and bundles where the quantity matters for buyer decisions.

#### Example mappings:

Schema Property	Magento Attribute
gtin13	ean
mpn	manufacturer_part_number
color	color
material	material
category	product_type
numberOfItems	number_of_items

## Configurable Products

---

Configurable products need special handling to show variant information.

### The Problem

A configurable product (e.g., "T-Shirt") has multiple simple products (e.g., "T-Shirt - Red - Small", "T-Shirt - Blue - Large"). Each variant may have different:

- Price
- Stock status
- SKU/GTIN

- Images

## The Solution: ProductGroup with Variants

Enable **Split Configurable Products Snippets** with **Use Variants** mode.

This generates:

```
{
  "@type": "ProductGroup",
  "name": "T-Shirt",
  "hasVariant": [
    {
      "@type": "Product",
      "name": "T-Shirt - Red - Small",
      "offers": { "price": "29.99" }
    },
    {
      "@type": "Product",
      "name": "T-Shirt - Blue - Large",
      "offers": { "price": "29.99" }
    }
  ]
}
```

## Configuration Options

Setting	Description
<b>Use reviews from parent product</b>	Apply parent's reviews to all variants (recommended)
<b>Use Parent Data for Simples</b>	Which attributes variants should inherit from parent

### Inherit from Parent:

- Name (if variants don't have unique names)
- Description (usually same for all variants)
- Brand (always same)
- Image (only if variants don't have unique images)

### Don't Inherit:

- Price (variants may have different prices)

- SKU/GTIN (always unique per variant)

## Bundle Products

---

Bundle products contain multiple items sold together.

### Enable Split Bundle

Set **Split Bundle Products Snippets** to **Yes** to generate schemas for each bundle item.

Setting	Description
<b>Use reviews from parent product</b>	Apply bundle reviews to items
<b>Use Parent Data for Simples</b>	Attributes to inherit from bundle

## Grouped Products

---

Grouped products display multiple related items.

### Enable Split Grouped

Set **Split Grouped Products Snippets** to **Yes** for individual item schemas.

Setting	Description
<b>Use reviews from parent product</b>	Apply grouped product reviews to items
<b>Use Parent Data for Simples</b>	Attributes to inherit from grouped parent

## Video Object

---

Include product videos in structured data.

Setting	Description
<b>Video Object Enable</b>	Include VideoObject schema for products with videos
<b>Video Object Source</b>	For configurables: use parent or child videos

### Requirements for VideoObject:

- Video must have a URL (YouTube, Vimeo, or direct link)
- Video should be associated with the product in Magento

## Additional Properties

---

Available since v2.10.0

Many products have attributes that matter to buyers but don't map to a standard Schema.org property. Think of specifications like "Weather Resistant", "UV Protection Factor", "Maximum Load Capacity", or "Seat Height". Without structured markup, search engines treat these as plain text and can't use them for filtering or rich results.

The Schema.org `additionalProperty` field solves this by wrapping custom attributes in `PropertyValue` objects. Google and other search engines can parse these as structured key-value pairs, making your product data more complete and machine-readable. This is especially valuable for categories where buyers compare technical specifications, such as furniture, outdoor equipment, electronics, and industrial products.

**Location:** Schema: Products → Additional Properties

### Configuration

Column	Description
<b>Product Attribute</b>	The Magento product attribute to read the value from
<b>Property Name</b>	The Schema.org property name shown in structured data (e.g., "Weather Resistant", "Seat Height")
<b>Value Type</b>	How to format the value: Text, Boolean, or Quantitative
<b>Unit Code</b>	UN/CEFACT unit code for quantitative values (e.g., CMT for centimeters)

### Value Types

Choose the right value type based on the nature of the attribute:

Type	Behavior	Best for
<b>Text</b>	Output as plain string	Materials, certifications, free-form specs
<b>Boolean</b>	Converts to "Yes" or "No"	Feature flags like "Waterproof", "Foldable", "Includes Cushions"

Type	Behavior	Best for
<b>Quantitative</b>	Outputs as <code>QuantitativeValue</code> with optional <code>unitCode</code>	Measurements like seat height, load capacity, cable length

**Tip:** For boolean attributes, the extension recognizes common truthy values ("yes", "true", "1", "ja", "oui") and falsy values ("no", "false", "0", "nee", "non") across multiple languages and normalizes them to "Yes" or "No".

## Example Output

A garden lounge set with custom specifications:

```
"additionalProperty": [
  {
    "@type": "PropertyValue",
    "name": "Weather Resistant",
    "value": "Yes"
  },
  {
    "@type": "PropertyValue",
    "name": "UV Resistant",
    "value": "Yes"
  },
  {
    "@type": "PropertyValue",
    "name": "Seat Height",
    "value": {
      "@type": "QuantitativeValue",
      "value": 45,
      "unitCode": "CMT"
    },
    "unitCode": "CMT"
  },
  {
    "@type": "PropertyValue",
    "name": "Maximum Load Capacity",
    "value": {
      "@type": "QuantitativeValue",
      "value": 150,
      "unitCode": "KGM"
    },
    "unitCode": "KGM"
  }
]
```

]

## Dimensions Mapping

---

Available since v2.10.0

Product dimensions are first-class properties in Schema.org's [Product](#) type. Google recognizes `width`, `height`, `depth`, and `weight` directly on a Product, and uses them for Google Shopping and product knowledge panels. Unlike additional properties, these don't need the `additionalProperty` wrapper — they are standard fields that search engines already know how to interpret.

The extension maps your Magento product attributes to these dimension properties using the `QuantitativeValue` format, which pairs a numeric value with a standardized [UN/CEFACT unit code](#). This ensures dimensions are unambiguous regardless of locale.

**Location:** Schema: Products → Dimensions Mapping

### Configuration

Column	Description
<b>Dimension</b>	The Schema.org property: <code>width</code> , <code>height</code> , <code>depth</code> , or <code>weight</code>
<b>Product Attribute</b>	The Magento attribute containing the numeric value
<b>Unit Code</b>	UN/CEFACT unit code (e.g., CMT for cm, MTR for m, KGM for kg)

**Important:** Make sure the attribute contains a numeric value only (e.g., "240" not "240 cm"). The extension extracts the number and pairs it with the configured unit code. If the attribute value contains text mixed with numbers, the extension will attempt to extract the numeric portion.

### Example Output

A table with width, height, and weight:

```
{
  "@type": "Product",
  "name": "Oak Dining Table",
  "width": {
    "@type": "QuantitativeValue",
    "value": 240,
    "unitCode": "CMT"
  },
}
```

```

"height": {
  "@type": "QuantitativeValue",
  "value": 75,
  "unitCode": "CMT"
},
"depth": {
  "@type": "QuantitativeValue",
  "value": 100,
  "unitCode": "CMT"
},
"weight": {
  "@type": "QuantitativeValue",
  "value": 45.5,
  "unitCode": "KGM"
}
}

```

## Common Unit Codes

Use [UN/CEFACT Common Codes](#) for standardized measurement units:

Code	Unit	Typical use
CMT	Centimeters	Furniture, home & garden
MTR	Meters	Large items, textiles
MMT	Millimeters	Precision/small items
INH	Inches	US-oriented stores
KGM	Kilograms	Weight (metric)
GRM	Grams	Lightweight items, jewelry
LBR	Pounds	Weight (US)

## Product Relationships

*Available since v2.10.0*

Search engines use product relationships to understand how items in your catalog connect to each other.

Schema.org defines two key properties for this: [isRelatedTo](#) for general product connections and [isAccessoryOrSparePartFor](#) for parts and accessories that belong to a main product.

Adding these relationships helps search engines build a richer product graph. This can improve how your products appear in "related items" and "people also viewed" features in search results, and it strengthens the overall topical authority of your product catalog by showing search engines the connections between your items.

**Location:** Schema: Products → Product Relationships

## Related Products (isRelatedTo)

Maps Magento's native product relationship types to Schema.org `isRelatedTo`. This tells search engines which products are related, similar, or complementary.

Setting	Description
<b>Enable</b>	Turn product relationships on or off
<b>Related Products (isRelatedTo)</b>	Which Magento relationship type to use: Related Products, Up-Sell Products, or Cross-Sell Products
<b>Related Products Limit</b>	Maximum number of products to include (default: 5). Higher values increase page size

### Which relationship type to choose:

- **Related Products** — Items frequently bought together or similar alternatives. Best general-purpose option.
- **Up-Sell Products** — Higher-end alternatives. Good for showing premium options.
- **Cross-Sell Products** — Complementary items (e.g., accessories shown at checkout). Good for connecting different product categories.

**Tip:** Keep the limit reasonable (3-5 products). Each related product adds to the JSON-LD payload size. Only enabled (visible) products are included — disabled products are automatically filtered out.

### Example output:

```
"isRelatedTo": [  
  {  
    "@type": "Product",  
    "@id": "https://store.com/matching-chair.html#product",  
    "name": "Matching Dining Chair",  
    "sku": "CHAIR-OAK-001",  
    "url": "https://store.com/matching-chair.html"  
  },  
  {  
    "@type": "Product",
```

```

    "@id": "https://store.com/table-extension.html#product",
    "name": "Table Extension Leaf",
    "sku": "EXT-OAK-001",
    "url": "https://store.com/table-extension.html"
  }
]

```

## Accessory / Spare Part For (isAccessoryOrSparePartFor)

Links a product to the main product it is an accessory or spare part for. This is a directional relationship — the current product *belongs to* a parent product, not the other way around.

Common use cases:

- Cushion sets for a lounge or sofa
- Replacement parts for machinery or appliances
- Phone cases linked to a specific phone model
- Ink cartridges linked to a printer

Setting	Description
<b>Accessory/Spare Part For</b>	Select an attribute containing the SKU or URL of the parent product

The attribute value can be either:

- A **product SKU** — the extension looks up the product and generates a full reference with name, SKU, and URL
- A **product URL** — used directly as the `@id` and `url` (useful for external or cross-store references)

**Example:** A cushion set that is an accessory for a lounge set.

```

"isAccessoryOrSparePartFor": {
  "@type": "Product",
  "@id": "https://store.com/premium-lounge-set.html#product",
  "name": "Premium Lounge Set",
  "sku": "LOUNGE-001",
  "url": "https://store.com/premium-lounge-set.html"
}

```

**Tip:** If a product referenced by SKU is not found or has been deleted, the property is silently omitted to avoid invalid structured data.

## Advanced Settings

---

### Hide Offer When Price is Zero

Enable to exclude the Offer schema when product price is €0.00. Useful for:

- "Price on request" products
- Free downloadable products
- Products with required options

### Custom Price Attribute

If you use a custom attribute for pricing (not Magento's standard price):

1. Enable **Use non-default attribute for price**
2. Select your custom price attribute

## Example Output

---

Complete Product schema output:

```
{
  "@context": "https://schema.org",
  "@type": "Product",
  "name": "Premium Wireless Headphones",
  "image": "https://store.com/media/headphones.jpg",
  "description": "High-quality wireless headphones with noise cancellation",
  "brand": {
    "@type": "Brand",
    "name": "AudioTech"
  },
  "sku": "AT-WH-001",
  "gtin13": "1234567890123",
  "offers": {
    "@type": "Offer",
    "url": "https://store.com/headphones",
    "price": "199.99",
    "priceCurrency": "EUR",
    "availability": "https://schema.org/InStock",
    "itemCondition": "https://schema.org/NewCondition",
    "priceValidUntil": "2025-12-31"
  }
}
```

```
  },
  "aggregateRating": {
    "@type": "AggregateRating",
    "ratingValue": "4.5",
    "reviewCount": "127"
  },
  "review": [
    {
      "@type": "Review",
      "author": { "@type": "Person", "name": "John D." },
      "reviewRating": { "@type": "Rating", "ratingValue": "5" },
      "reviewBody": "Excellent sound quality!"
    }
  ]
}
```

## Validation Checklist

---

After configuration, validate your Product schema:

- Use [Rich Results Test](#)
- Check for errors (red) - must fix
- Check for warnings (yellow) - recommended to fix
- Verify all expected fields appear

### Common issues:

- Missing brand → Map brand attribute
- Missing priceValidUntil → Configure price validity
- Image too small → Use larger product images

## 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 Rich Snippets Suite extension on [magmodules.eu](http://magmodules.eu)

# All articles for Rich Snippets Suite

---

## Installation

---

1	<a href="#">Installation using Composer (recommended)</a>
2	<a href="#">Installation using the Adobe Marketplace</a>
3	<a href="#">Install through FTP and SSH</a>

## Configuration

---

1	<a href="#">FAQ Widget</a>
2	<a href="#">Quick Start Guide</a>
3	<a href="#">Configuration</a>
<b>4</b>	<b><a href="#">Product Schema (current)</a></b>
5	<a href="#">Organization &amp; Local Business</a>
6	<a href="#">Shipping &amp; Returns</a>
7	<a href="#">Social Markup</a>

## Troubleshooting

---

1	<a href="#">Google Validation</a>
2	<a href="#">Rich Results Visibility in Google</a>
3	<a href="#">Troubleshooting</a>

## Background

---

1	<a href="#">Review Sources</a>
2	<a href="#">About Structured Data</a>
3	<a href="#">CLI Commands</a>

