

CSV Product Upload Guideline: Detailed Column Documentation

Column: **Name**

- **Required:** Yes
- **Data Type:** String
- **Description:**
This is the primary display name of the product. It is shown to users throughout the storefront, including product listings, detail pages, and during checkout.
- **Validation Rules:**
 - Must not be empty.
 - Recommended to keep it under 100 characters for UI consistency.
- **Best Practices:**
 - Include key information (e.g., size or model) for uniqueness.
 - Example: "2025 Eco-Friendly Bamboo Toothbrush"

Column: **SKU**

- **Required:** No
- **Data Type:** String
- **Description:**
Stock Keeping Unit – a unique code assigned to each product for inventory management.
- **Validation Rules:**
 - Should be unique if provided.
 - Alphanumeric codes are commonly used.
- **Best Practices:**
 - Use a consistent format for easy tracking (e.g., TSHIRT-XL-BLACK).
 - Example: "BAMBOO-TS-2025"

Column: **Description**

- **Required:** No
- **Data Type:** String (HTML supported)
- **Description:**
Full description with details and benefits of the product. Can contain formatted content.
- **Best Practices:**
 - Write in paragraphs or use bullet points.
 - Use simple HTML if needed.

Column: **Short Description**

- **Required:** No
- **Data Type:** String
- **Description:**
A brief summary shown on listings or previews.
- **Best Practices:**
 - Limit to ~200 characters.
 - Example: "Soft and breathable cotton t-shirt for daily wear."

Column: **Active**

- **Required:** Yes
- **Data Type:** Boolean (1 or 0)
- **Description:**
Indicates whether the product is active and visible on the storefront.
- **Validation Rules:**
 - 1 = Published, 0 = Hidden

Column: Brand

- **Required:** No
- **Data Type:** String
- **Description:** The brand name associated with the product. If the brand doesn't exist in the system, it should be created during the import process.
- **Validation Rules:**
 - Brand names should be unique and properly capitalized.
- **Best Practices:**
 - Use consistent brand naming across all products.
 - Avoid trailing or leading spaces.
- **Example:**
 - "Nok1a"
 - "Tony"
- **System Behavior:**
 - During CSV import, the system will check if the brand exists. If not, it will automatically create it.

Column: Categories

- **Required:** No
- **Data Type:** String (category path separated by ///, multiple entries comma-separated)
- **Description:** Defines the full hierarchical path of categories the product belongs to. Use /// to represent parent-child relationships.
- **Example:**
 - "Men's Fashion///Clothing///Jeans"
 - For multiple categories: "Electronics///Accessories,Men's Fashion///Clothing///Jeans"
- **Processing:**
 - Each category level is split and mapped to a parent-child relationship.
 - System must create parent categories if they don't exist.

Column: **Tax Class**

- **Required:** No
- **Data Type:** Integer (ID of tax class)
- **Description:** Specifies the applicable tax class for the product. Useful for determining VAT/GST rules based on region or product category.
- **Validation Rules:**
 - Must match an existing tax class ID.
 - Leave blank if tax is not applied.
- **Best Practices:**
 - Ensure tax classes are predefined in the system.

Example:

- 1 (Shipping Address)
- 2 (Standard Rate)
- 3 (Reduced Rate)

Column: **Tags**

- **Required:** No
- **Data Type:** String (comma-separated)
- **Description:**
Keywords that help users search or filter products.
- **Example:**
 - "eco-friendly,bamboo,oral care"

Column: **Price**

- **Required:** Yes
- **Data Type:** Decimal
- **Description:**
The base selling price of the product before any discounts.
- **Validation Rules:**
 - Non-negative decimal number.
 - Use period (.) as the decimal separator.
- **Best Practices:**
 - Avoid currency symbols.
 - Example: "19.99"

Column: **Special Price**

- **Required:** No
- **Data Type:** Decimal
- **Description:**
A discounted price is offered for a limited time.
- **Validation Rules:**
 - Must be less than the **price** value.
 - Use only when a discount is active.
- **Best Practices:**
 - Example: "14.99" if the **price** is 19.99

Column: **Special Price Type**

- **Required:** No
- **Data Type:** Enum (**fixed** or **percent**)
- **Description:**
Determines whether the **special_price** is a fixed amount or a percentage discount.
- **Validation Rules:**
 - Must be either **fixed** or **percent**
- **Example:**
 - **fixed:** Sale price is **14.99**
 - **percent:** If 20, then 20% off the base price

Column: **Special Price Start**

- **Required:** No
- **Data Type:** Date (**YYYY-MM-DD**)
- **Description:**
Start date for the special price period.
- **Validation Rules:**
 - Must follow date format.
- **Example:**
 - **"2025-06-01"**

Column: **Special Price End**

- **Required:** No
- **Data Type:** Date (YYYY-MM-DD)
- **Description:**
End date for the special price period.
- **Validation Rules:**
 - Must follow the date format.
 - Must be later than `special_price_start`
- **Example:**
 - `"2025-06-10"`

Column: **Manage Stock**

- **Required:** No
- **Data Type:** Boolean (1 or 0)
- **Description:**
Indicates whether the system should track product stock quantity.
- **Validation Rules:**
 - 1 = Yes, 0 = No
- **Best Practices:**
 - Set to 1 if `qty` is used.

Column: **Quantity**

- **Required:** No
- **Data Type:** Integer
- **Description:**
The number of available units in stock. Used only if **manage_stock = 1**.
- **Validation Rules:**
 - Must be a non-negative integer.
- **Example:**
 - "100" means 100 units in inventory.

Column: **In Stock**

- **Required:** No
- **Data Type:** Boolean (1 or 0)
- **Description:**
Shows whether the product is available for purchase.
- **Validation Rules:**
 - 1 = In stock, 0 = Out of stock
- **Example:**
 - "1" if available to order.

Column: **Base Image**

Required: No

Data Type: String (URL or local path)

Description:

This is the **main display image** for the product. It appears as the **primary visual** in listings, product detail pages, and featured areas.

Validation Rules:

- If the image is provided as a **URL**:
 - It must be valid and publicly accessible.
- If the image is a **local filename** (e.g., `img1.jpg` or `images/img1.jpg`):
 - The file must be included in the **ZIP file** that is uploaded alongside the CSV.
 - The path must exactly match the file structure inside the ZIP archive.

System Behavior:

- When the system processes the CSV, it **matches the image name** with files inside the uploaded ZIP archive.
- If the path is:
 - `img1.jpg`: This means the image must be located in the **root directory** of the ZIP.
 - `images/img1.jpg`: The image must be inside the **images/ folder** inside the ZIP.
- On a successful match, the image is extracted and saved in the media library or product images directory.

Example:

- `main-product.jpg` → located directly in the root of the ZIP
- `products/shoes/front.jpg` → must exist as `products/shoes/front.jpg` in the ZIP

Column: **Additional Images**

Required: No

Data Type: String (comma-separated)

Description:

These are **extra images** that appear in the **product image gallery**, allowing users to view the product from different angles or in different variations.

Validation Rules:

- Each image name must be:
 - A valid path as described above (`img2.jpg`, `gallery/img3.jpg`, etc.)
 - Separated by commas, no spaces between values.

System Behavior:

- Each image listed is searched within the uploaded ZIP using the **exact path**.
- If any image is **not found**, the system logs an error or skips that particular image.
- All valid images are added to the product gallery in the order listed.

Example:

- `img2.jpg`, `img3.jpg` → both located in ZIP root
 - `gallery/view1.jpg`, `gallery/view2.jpg` → both must exist inside the `gallery/` folder in ZIP
-

How the ZIP File Works

Image Upload Flow:

1. **You upload a ZIP file** containing all product images.
2. The system reads the **image filenames in the CSV** (Base and Additional) and looks for those files inside the ZIP archive.
3. It **preserves folder paths**, so structure **matters**.
 - `img1.jpg` → root
 - `folder/img1.jpg` → must exist exactly as `folder/img1.jpg`

ZIP File Best Practices:

- Always **match filenames and paths** between the CSV and ZIP exactly.
- Avoid spaces or special characters in filenames.
- Keep images **well organized** in folders if needed (e.g., `main/`, `gallery/`).

Good Example:

CSV:

Base Image: `main/front.jpg`

Additional Images: `gallery/img1.jpg`, `gallery/img2.jpg`

ZIP Structure:

- `main/`
 - `front.jpg`
- `gallery/`
 - `img1.jpg`
 - `img2.jpg`

Product Attribute String Format – Quick Guide for Excel Import

Use this guide to correctly format your product attributes for importing via Excel.



Format Structure

Each attribute should follow this pattern:

- **[Group] Attribute Name | Categories: Category1, Category2 | Slug: slug_name | Filterable: true/false | Values: value1, value2, value3**

Separate multiple attributes using double pipes **||**.



Example Explained

- **[Hardware] Battery | Categories: Mobile///SmartPhone, Mobile///Android, Mobile///iPhone | Slug: battery | Filterable: true | Values: 3000mAh, 4000mAh, 5000mAh**
- **[Hardware]** → Group name for organization (e.g., Hardware)
- **Battery** → Attribute name (what the attribute is)
- **Categories** → Product categories affected, separated by commas
 - Use **///** for category levels:
 - **Parent///Child///Sub-child**
 - Example: **Mobile///SmartPhone** means "SmartPhone" under "Mobile"
- **Slug** → A unique, lowercase, no-space identifier (e.g., battery)
- **Filterable** → Set to **true** if it should be a filter on the site
- **Values** → Allowed values (comma-separated)

Best Practices

- Always separate multiple attributes with `| |`
- Ensure that category levels use `///`
- Keep slugs clean (lowercase, no special characters or spaces)
- Double-check that values are properly comma-separated
- Avoid typos to prevent import issues



Sample with Multiple Attributes

- `[Hardware] Battery | Categories: Mobile///SmartPhone | Slug: battery | Filterable: true | Values: 3000mAh, 4000mAh || [Hardware] Display Size | Categories: Mobile///SmartPhone | Slug: display_size | Filterable: true | Values: 5.5", 6.1"`

Feel free to use this as a reference while building your Excel import sheet!

Option String Format Guideline for Excel Upload

Each product can include multiple configurable options such as size, personalization, gift wrap, etc. These options must be formatted correctly in the "Options" column of your Excel file.

General Format

- Each option is separated by: ||
- Each property within an option is separated by: ;

Example:

```
name=Size;type=dropdown;is_required=1;values[0][label]=Small;values[0][price]=0.00;values[0][price_type]=fixed||name=Gift Wrap;type=checkbox;is_required=0;values[0][label]=Add Gift Wrap;values[0][price]=4.50;values[0][price_type]=fixed
```

Supported Option Types

1. Dropdown, Checkbox, Radio

- Format:

2. name=...;type=dropdown|checkbox|radio;is_required=0|1;values[0][label]=...;values[0][price]=...;values[0][price_type]=fixed

- Example:

3. name=Size;type=dropdown;is_required=1;values[0][label]=Small;values[0][price]=0.00;values[0][price_type]=fixed;values[1][label]=Large;values[1][price]=2.00;values[1][price_type]=fixed

4. Checkbox Custom

- Use for multiple selectable options with prices.
- Format:

5. name=...;type=checkbox_custom;is_required=0|1;values[0][label]=...;values[0][price]=...;values[0][price_type]=fixed

- Example:

6. name=Dietary

Restrictions;type=checkbox_custom;is_required=0;values[0][label]=Vegan;values[0][price]=1.00;values[0][price_type]=fixed

7. Field (Single-line Text)

- Format:

8. name=...;type=field;is_required=0|1;price=...;price_type=fixed;max_characters=...

- Example:

9. name=Personalization;type=field;is_required=0;price=5.00;price_type=fixed;max_characters=15

10. Textarea (Multi-line Text)

- Format:

11. name=...;type=textarea;is_required=0|1;price=...;price_type=fixed;max_characters=...

- Example:

12. name=Personalized

Note;type=textarea;is_required=0;price=2.00;price_type=fixed;max_characters=250

13. Date Picker

- Format:

14. name=...;type=date;is_required=0|1;price=...;price_type=fixed

- Example:

15. name=Delivery Date;type=date;is_required=1;price=0.00;price_type=fixed

Important Rules

- No leading or trailing spaces around || or ;
- Required fields: name, type, is_required
- If type uses values[], each value must have: label, price, and price_type
- price_type must always be fixed
- is_required should be either 0 (optional) or 1 (required)

Sample Complete Option String

name=Size;type=dropdown;is_required=1;values[0][label]=Medium;values[0][price]=0.00;values[0][price_type]=fixed;values[1][label]=Large;values[1][price]=2.00;values[1][price_type]=fixed||name=Personalization;type=field;is_required=0;price=5.00;price_type=fixed;max_characters=15||name=Gift Wrap;type=checkbox;is_required=0;values[0][label]=Add Gift Wrap;values[0][price]=4.50;values[0][price_type]=fixed