

# Product Catalog Specification

Monetate's product catalog specification is loosely based on [Google's product data specification](#).

## Catalog Count

You can upload as many product catalogs as you want under a single account as long as they share product IDs (`item_group_id`) and categories (`product_type`) across each one. All categories across catalogs within an account must be in the same language. If you share a single tag across multiple domains (for example, `multi-brand`), then you must supply a single product catalog.

## Required Attributes

This table contains all the columns, or attributes, that must appear in a product catalog file that you upload along with the type of data it must contain.

If the column header text in the CSV file you upload does not exactly match how the attribute appears in this table, then Monetate cannot process the file. Furthermore, you must provide a value for each required attribute in each row of the file.

Attribute	Data Type	Example	Description
<code>item_group_id</code>	String	<code>AB12345</code>	Identifier for a group of products that come in different versions; the value of this attribute <i>must</i> match the product ID (PID) passed in your integration  <b>NOTE:</b> This field can contain a maximum of 50 characters.
<code>id</code>	String	<code>A2B4</code>	A product's unique identifier; the value of this attribute <i>must</i> match the product SKU if you're passing SKUs in your integration  <b>NOTE:</b> This field can contain a maximum of 50 characters.
<code>title</code>	String	<code>Men's Pique Polo Shirt</code>	A product's name, which appears in the recommendations

Attribute	Data Type	Example	Description
image_link	String	http://www.example.com/image1.jpg	The URL for a product's main image
link	String	http://www.example.com/asp/sp.asp?cat=12&id=1030	The URL for a product's landing page
description	String	"Red, 100% cotton, large men's t-shirt"	A product's description
price	Number	15.98	A product's price
product_type	Comma-separated string	"Clothing, Women, Sale"	Separate each product type with a comma, and then place a double quotation mark at the beginning and at the end of the entire string to escape it. A product type that contains a comma must itself be surrounded by escaped quotes. See <a href="#">Comma Escaping</a> within this documentation for more information.

## Optional Attributes

This table contains the attributes you can include in addition to the required attributes in a product catalog file that you upload, along with the type of data each optional attribute must contain.

If the column header text in the CSV file you upload does not exactly match how the attribute appears in this table, then Monetate cannot process the file.

Attribute	Data Type	Example	Description
additional_image_link	String	http://www.example.com/image2.jpg	The URL for an additional image for a product
adult	Boolean	yes	Indicates if a product contains nudity or sexually suggestive content
age_group	String	infant	A product's target demographic

Attribute	Data Type	Example	Description
availability	String	in stock	A product's availability
availability_date	Datetime	2022-09-30T12:10:45.000145Z	<p>The date a preordered product becomes available for delivery</p> <p><b>NOTE:</b> You must include this field in any product catalog that you use with the Newest premium recommendation algorithm. See <a href="#">Recommendation Algorithms</a> for more information.</p>
brand	String	MyBrand	A product's brand name
color	String	red	A product's color
condition	String	refurbished	A product's condition
energy_efficiency_class	String	A+	A product's energy label
expiration_date	Datetime	2022-09-30T12:10:45.000145Z	The date on which a product should stop appearing
gender	String	unisex	A product's target gender

Attribute	Data Type	Example	Description
google_product_category	String or number	Apparel & Accessories > Clothing > Dresses or 2271	<p>The full category path or numerical category ID for a product as defined in the Google product taxonomy</p> <p><b>NOTE:</b> Including this attribute and a supported value enables advanced analysis of the product catalog data. If you don't use a Google-defined value, then it's ignored. See <a href="#">Google's definition of this attribute</a> for the supported values.</p>
is_bundle	Boolean	yes	Indicates if a product contains multiple different products featuring one main product
loyalty_points	String	100	The number and type of loyalty points a customer receives when buying a product
material	String	silk	A product's fabric or material

Attribute	Data Type	Example	Description
mobile_link	String	http://m.example.com/asp/sp.asp?cat=12&id=1030	A product's mobile-optimized landing page when you have a different URL for mobile and desktop traffic
mpn	String	MO12345NETATE	A product's manufacturer part number (MPN) as assigned by a manufacturer
multipack	Number	6	The number of identical products sold within a multipack
pattern	String	striped	A product's pattern or graphic print
promotion_id	String	BOGO22	An identifier that allows to you match products to promotions
quantity	Number	42	The inventory count of a product
sale_price	Number	14.49	A product's sale price
sale_price_effective_date_begin	Datetime	2022-09-30T12:10:45.000145Z	The date on which a product's sale price, defined in sale_price, starts
sale_price_effective_date_end	Datetime	2022-10-30T12:10:45.000145Z	The date on which a product's sale price, defined in sale_price, ends

Attribute	Data Type	Example	Description
shipping	String	5.99	A product's shipping cost
shipping_label	String	perishable	A label assigned to a product for calculating shipping costs
shipping_height	String	20 in	A product's height used to calculate the shipping cost by dimensional weight, as measured in inches (in) or centimeters (cm)
shipping_length	String	20 in	A product's length used to calculate the shipping cost by dimensional weight, as measured in inches (in) or centimeters (cm)
shipping_width	String	20 in	A product's width used to calculate the shipping cost by dimensional weight, as measured in inches (in) or centimeters (cm)
shipping_weight	String	3 lb	A product's weight used to calculate the shipping cost by dimensional weight, as measured in imperial units (lb or oz) or metric units (g or kg)

Attribute	Data Type	Example	Description
size	String	XL	A product's size
size_type	String	maternity	A product's cut
tax	String	US:CA:5.75:y	A product's sales tax rate in percent (5.75), with the optional subattributes of country (US), region or state (CA or postal code, or Boolean value of tax charged on shipping (y) appended using a colon between each

## Custom Fields

You can include data not defined in the Monetate product catalog specification. For example, you may want to send additional data to enhance [recommendation strategies](#) with different filters or to render recommendations with additional information, such as ratings. You can submit this data by including custom fields, or columns, in a product catalog file.

You must define each custom field and the type of data it contains when you [create the product catalog schema](#).

You can use comma-separated values in a custom field by selecting **Comma Separated String** for the field when creating the schema. Monetate then considers the value within the field as a list of values that are separated by commas. Fields with this data type are subject to additional rules as described in the [Escaping Comma-Separated Strings](#) section within this documentation.

## Custom Fields in the User Interface

Custom field headers don't appear in the dataset attribute search section of the platform at this time. Custom fields are available when you're setting up recommendations and adding filters.

## File Format

A product catalog file must be in the CSV or TSV format.

When uploading product catalog files [via SFTP](#), you can compress them and upload that resulting file if it's in the ZIP (.zip) or Gzip (.gz) format. You only use the .gz file extension for a Gzip file. Monetate doesn't accept files with the .gzip file extension.

Each field must have a header.

The file must use UTF-8 character encoding without a byte order mark (BOM).

Download this CSV file to view an example of a product catalog file with all the required attributes: [Sample Product Catalog \(CSV\)](#).

You can use this sample as a template for your own product catalog files.

## Comma Escaping

For a product catalog file to pass validation and for the platform to correctly parse a value that contains a comma and to correctly parse comma-separated strings, you must use comma escaping, which involves using sets of double quotation marks to define individual values. You must also escape quotation marks that are natively part of a value.

The escaping methods described here best ensure that Monetate, because of the way it processes CSV files, correctly parses comma-separated strings.

## Escaping Single Values

If a value for an attribute includes a comma, then the value must be escaped by placing opening double quotation marks at the beginning of the value and closing double quotation marks at the end. The quotation marks are necessary to signal that the comma within the value *is not* a separator.

Consider this example. A CSV file contains the value Philadelphia, PA for an attribute, but that value isn't contained—escaped—within opening and closing double quotation marks. When the file is processed, Philadelphia is treated as one value and PA is treated as another. Therefore, the correct way to submit the value is "Philadelphia, PA".

If a value contains double quotation marks that are natively part of the value, then you must place the entire value in double quotation marks *and* escape each quotation mark inside the value by doubling it. For example, a product's description is Feed Me Right "Meow" ceramic cat food bowl. When you submit this information as a value in the description attribute, you must put the entire text in double quotation marks, and then double the quotation marks that are inside the value: "Feed Me Right ""Meow"" ceramic cat food bowl". If you don't escape the quotation marks within the description and escape the entire description, then the value won't parse correctly.

## Escaping Comma-Separated Strings

When the platform processes a CSV file, it parses the data twice. In the first round, it separates the file into cells.



In the second round, it separates cells marked as containing comma-separated strings into subvalues. For this reason, you must put two sets of quotation marks around a value that contains a comma when it's in a comma-separated string and then put the full string in quotation marks.

These escaping rules for comma-separated strings apply to TSV files as well as to CSV files.

Consider this example. You want to define a product with the following three product types:

- Office Supplies > Mailing, Packing & Shipping > Packing Materials > Packing Papers
- Bulk Products
- Seasonal Discounted Offerings > December Discounts

For the platform to correctly interpret the three product types in this comma-separated string as a single value, you must do the following:

1. Place two sets of quotation marks around the first product type: `"" Office Supplies > Mailing, Packing & Shipping > Packing Materials > Packing Papers""`
2. Put the entire string in quotation marks so it's considered one value: `""""Office Supplies > Mailing, Packing & Shipping > Packing Materials > Packing Papers""",Bulk Products,Seasonal Discounted Offerings > December Discounts"`

Even if only one product type appears in the comma-separated string, if it includes a comma, then you must put two sets of quotation marks to signify the presence of the comma *and* apply a third set of quotation marks around the whole string.

For example, the product type is Office Supplies > Mailing, Packing & Shipping > Packing Materials > Packing Papers. For it to be parsed correctly, the string must appear in the file as follows: `""""Office Supplies > Mailing, Packing & Shipping > Packing Materials > Packing Papers""""`.

The same rules apply if the string natively includes a double quotation mark or marks. You place two sets of quotation marks around the text with the native quotation mark, and then you place the entire string in quotation marks.

For example, you once again have three product types:

- Bulk Products
- Seasonal Discounted Offerings > December
- 4" Steel Fasteners

For these three product types to be correctly parsed as one value, you must first put two sets of double quotation marks around the text indicating the fasteners are 4 inches in size, so `""4""`. With the quotation mark properly escaped, you must then escape the entire string: `""Bulk Products,Seasonal Discounted Offerings > December, ""4""Steel Fasteners"`.

If the comma-separated string were to contain only the product type 4" Fasteners, Steel, the correct way to escape this string so that the platform parses it correctly is as follows: `""4""Fasteners, Steel""`.

The platform's parsing rules for comma and quotation marks are the same as the standards set forth in [RFC](#)

4180: Common Format and MIME Type for Comma-Separated Values (CSV) Files published by the Internet Engineering Task Force (IETF). Refer to this publication for the exact definitions and standards.