

# Product Data Feed Requirements



As mentioned on the page [Prepare the Product Data](#), this part of the documentation describes the requirements that your product data feed needs to fulfill in order to **build Product Guides entirely yourself** with the excentos Workbench.

If you have more complex feeds (e.g. web services or XML feed), please return to the [Prepare the Product Data](#) section or [contact excentos support](#) if you have any questions regarding your data - we'll help you cover any data feed requirements.

Table of Contents:

- [Upload Requirements and Limitations](#)
- [Data Feed File Requirements](#)
- [Required Fields and Their Names](#)
  - [General Data Attributes](#)
  - [Product Data Sources](#)
  - [Category-specific Product Data](#)
- [Data Feed for international rollouts](#)
- [Mime Type Detection](#)

## Upload Requirements and Limitations

- The file has to be a **CSV** (comma-separated values) file. Only this file type is accepted.
- If your product data is written in an Excel sheet (.xls or .xlsx) or in an OpenDocument format (.ods) which are used by LibreOffice, the file has to be exported to a CSV file (see [Export the Product Data File into CSV](#)).

## Data Feed File Requirements

If you want to **connect your product data directly to the Workbench**, the product data CSV file needs to fulfill the following requirements (as mentioned above, please contact excentos for more complex formats such as XML or JSON):

Type / Field	Specification
File type	CSV file (preferred encoded is <b>UTF-8</b> )
Field separator	Needs to be one of the following: ',' (comma), ';' (semi-colon), ':' (colon), ' ' (pipe), tab or whitespace character.
Text delimiter	Text cells have to be surrounded by '"' (double quote) or "'" (single quote). One of them at the beginning of a cell; one at the end of a cell. This is especially important if the text contains a specified field separator.
Escape character	The escape character that is used within the text cells. Normally this is a '\' (backslash), to escape a single double quote, another single double quote can also be used.
Decimal separator	Default is a '.' (dot), e.g. '799.99'. For comma separated decimals (e.g. '799,99') you need to set the number format in the Workbench.
Multiple values separator	<p>If some attributes contain cells with <b>more than one value in a cell</b> (e.g. the attribute color with the values black and blue), these values need to be separated by a separator (e.g. with a <i>comma</i> such as "black,blue"). Otherwise these two values will be interpreted as one value and cannot be split by using the "Multi Value" switch button in the data attribute table (see <a href="#">Upload your Product Data File</a>, <i>Multi Values</i>).</p> <div> You can change the global <i>multi values separator</i> anytime as well as the attribute-specific separator in the <a href="#">Product Data</a> view.</div>
Units	<p>Such as kilogram, currencies etc. <i>should be in a separate field</i> if the data field is required to be interpreted as a number.</p> <p>(this is a <i>suggestion</i> only because the unit is only required if it shall be displayed in the Product Guide in the result list)</p>



#### Hint

The *field separator* and *text delimiter* as well as the *file type encoding* and the *escape character* can be set during the [update process of your product data file](#).

## Required Fields and Their Names

The Workbench **expects a fixed name convention** for some *core product attributes*.

- The **"Required"** attributes in the table below are the minimal set of data that every Product Guide needs to work properly.
- The **"Recommended"** attributes are technically optional, but it is recommended you use them in order to display certain information like the product name or price in the [product recommendation card](#).



#### Attribute Mapping

Despite that fixed naming convention it is fortunately possible to **map** the underlying attribute names to your product attribute naming schema (see [Attribute Mapping](#)).



#### Hint

If you build up your product data from scratch, the easiest way to fulfill the data requirements is to start with the [Template Data File](#).

## General Data Attributes

*General product attributes* are identical across all product categories of your Shop / Website.

The **required** and **recommended data attributes** and the name conventions for the data fields, which can come from [several data sources](#) are:

Attribute Name	Required	Recommended	Remarks	Visible in Recommendation
id	X	-	Unique ID for a single product <div> If your feed does not contain that kind of information, an internal generic id is created to distinguish all products. This internal id is not visible in the data or the Workbench at all.</div>	-
name	(X)	X	Name of the product <div> The <i>name</i> is not technically required, but makes it a lot easier for you and the end-users to identify products in the result list.</div>	✓
image	(X)	X	Product image URL, e.g. a URL to the product image on your website <div> The <i>image</i> is not technically required, but makes it a lot easier for you and the end-users to identify products in the result list.</div>	✓
price	-	X	Product price (decimal number)	✓
manufacturer	-	X	Manufacturer or brand name	✓
detailUrl	-	X	URL to the product details page	✓
variantId	-	-	Master variant ID of a product. Useful if your product assortment contains e.g. color or size variants. excentos can collect them to make sure only the master product is displayed in the product result list.	-
description	-	X	Product description text	✓



### All attribute names are case sensitive

This means that the name 'price' and 'Price' will be recognized as two different attributes (with lowercase and uppercase 'p'). Please make sure all attributes are **written correctly** to match the required/recommended attributes above.

Otherwise make sure your own attribute names are **mapped correctly**. Else the corresponding information cannot be displayed or executed properly.



For more information on possible data attributes and which data might help to even improve the Product Guide, please also check <https://www.excentos.com/en/documentation/data-feed-requirements>.

## Product Data Sources



This part of the documentation describes different Product Data sources. The Product Data consists of several data sources and can also be connected by different connector types (HTTP, FTP, SFTP).

excentos can help to preprocess these Product Data sources to use them for your Product Guide. [Contact excentos support](#) if you have any questions regarding your Product Data sources.

## Possible Product Data sources

Source	Description	Recommended	Optional
PIM	product information like part numbers, variant id, descriptions etc.	x	
technical attributes	characteristics of a product; can be contained in master PIM or in another database	x	
availability feed	defines product availability per region		x
country-specific product definitions	e.g. different certificates per product; can be contained in master PIM or in other database		x

## Category-specific Product Data

Category-Specific Attributes describe the product itself, such as technical data, possible use-cases and everything that distinguishes one product from the other. For every Product Guide *all the category specific attributes* are needed, like e.g. the screen resolution of a notebook computer, the shoe size of a mountaineering boot, the range of height adjustment of an office desk etc. The more data there is, the more powerful and detailed the Product Guide can be.

The category-specific attributes can have the same data structure (mostly the case with CSV files) or a separate data structure / different attributes per category (mostly the case with XML data structures).

Category-Specific Attributes are *not required* for the data sourcing option „third-party licensed data“.

## Example of category-specific attributes

Typical CSV file with product data for, e.g., bicycles:

[blocked URL](#)

This is how the CSV file needs to look like if you open it in e.g. Excel if you have correctly set the text delimiters (in the example above a *comma* is used):

[blocked URL](#)

## Data Feed for international rollouts

You can either provide

- **individual feeds per market** / country / etc.
- or tell excentos a **filter criterion** within a central feed from which excentos can decide in which market to include the product.

If prices, product names and descriptions differ, it is recommended to provide separate data feeds for each market (although it would be technically possible to contain all markets in one feed).

## Mime Type Detection

---

During the upload process the Workbench is analyzing the data file type. Only if the file is a valid CSV the file can be uploaded. The corresponding **mime type** (internal format) for CSV files is **'text/csv'**. Unfortunately, there are 600+ words, word phrases or hexadecimal values (mostly at the start of the file) that can affect the mime type. That means that a word like **'Article'** at the beginning of a CSV file results in the mime type **'message/news'**. With that resulting type, it is not possible to upload the CSV anymore although the file is completely valid. Because of the huge amount of words and word phrases, excentos is not able to prevent these errors during the upload process. Nevertheless, the major part of these words/phrases are not relevant for a CSV file use case, like detecting python or image files.

If you have any problems regarding this issue do not hesitate to [contact us](#). Some "bad words" examples are listed below:

Bad words	Mime Type
Article	message/news
Path:	message/news
[version]	application/inf
StartFontMetrics	application/x-font-adobe-metric
Rar!	application/x-rar-compressed
Spreadsheet	application/x-sc
FWS	application/x-shockwave-flash
CWS	application/x-shockwave-flash
PSID	audio/prs.sid
BEGMF	image/cgm
icns	image/icns
AGD2	image/x-freehand
FreeHand10	image/x-freehand
MATLAB	application/x-matlab-data
#!/bin/perl	text/x-perl
#!/bin/python	text/x-python
FLV	video/x-flv
MOVI	video/x-sgi-movie
\037\213	application/gzip

[Mime Type XML](#)

Read about the [Template Data File](#) and [Example Data File](#) to get started building your own Data File