

Table of Contents

How to use Data Feed Manager?

STEP 1: Configure Data Feed Manager in a general way

STEP 2: Configure your data feeds

Export new template

Import new template

Create a template

Configuration

Categories

Filters

Scheduled Tasks

FTP Settings

STEP 3: Generate the data feeds

Generate the data feeds manually

Generate the data feeds automatically through the cron tasks

Manage your data feed templates thanks to Data Feed Manager

How to generate XML files with Data Feed Manager

How to generate CSV/TXT files with Data Feed Manager?

Attribute management in the Data Feed Manager templates

Attribute specifications

PARENT ATTRIBUTES

{attribute_code parent}

{attribute_code configurable}

{attribute_code grouped}

{attribute_code bundle}

PRODUCT IDENTIFIERS

{id}

{G:ID}

{G:ITEM_GROUP_ID}

{sku}

{brand}

{manufacturer}

BASIC INFORMATION

{inc}

{name}

{G:TITLE}

{description}

{G:DESCRIPTION}

{short_description}

{condition}

{weight}

PRICES AND PROMOTIONS

{price}
{G:PRICE}
{normal_price}
{special_price}
{price_rules}
{group_price}
{G:SALE_PRICE}
{is_special_price}
{tier_price}
{min_price}
{max_price}
{G:PROMOTION_ID}

LINKS

{url}
{G:LINK}
{image}
{G:IMAGE_LINK}
{host}
{url_key}

CATEGORIES

{categories}
{G:PRODUCT_TYPE}
{category_mapping}
{G:GOOGLE_PRODUCT_CATEGORY}
{attribute_set}

AVAILABILITY

{status}
{stock_status}
{G:AVAILABILITY}
{is_in_stock}
{qty}
{min_sale_qty}

REVIEWS

{review_count}
{review_average}
{G:PRODUCT_REVIEW}

PRODUCT VARIANTS

{gender}
{age_group}
{color}
{size}
{material}
{pattern}

BASIC OPTIONS

{attribute_code,[strip_tags]}

{attribute_code,[cleaner]}

{attribute_code,[substr],[max_length],[end_string]}

{attribute_code,[strtoupper]}

{attribute_code,[strtolower]}

{attribute_code,[inline]}

{attribute_code,[html_entity_decode]}

{attribute_code,[htmlentities]}

{attribute_code,[implode],[separating_string]}

{attribute_code,[float],[decimals]}

STOCK IN THE CHANNEL USERS

{SC:URL}

{SC:EAN}

{SC:IMAGES}

{SC:DESCRIPTION}

PHP API for Data Feed Manager

API coding standards

Attribute calls

PHP functions in attribute calls

Conditional values

Opening and closing tags

Outputting a value

Variables scope

API Objects, Variables and Methods

Object \$product

Object \$this

String | Integer | Boolean | Array \$self

Variable \$param

Custom attributes use

Custom options use

Shopping engines configurations

How to configure my template with Google Shopping?

Usefull links for Google Shopping

File format with Google Shopping

Required attributes with Google Shopping

Recommended attributes with Google Shopping

Example of template with XML format with Google Shopping

How to create a data feed in Google Merchant Center?

How to configure my template with Shopping.com?

File format for Shopping.com

Required attributes for Shopping.com

Recommended attributes for Shopping.com

Example of template for Shopping.com with XML format

How to configure my template with Bing Shopping?

File format for Bing Shopping

Required attributes for Bing Shopping

Recommended attributes for Bing Shopping

Example of template for Bing Shopping with TXT format

How to configure my template with Shopzilla?

File format for Shopzilla

Required attributes for Shopzilla

Recommended attributes for Shopzilla

Example of template for Shopzilla with TXT format

How to configure my template with Amazon?

File format with Amazon

Informations groups with Amazon

Required attributes with Amazon

Recommended attributes with Amazon

How to configure my template with Price Grabber?

File format with Price Grabber

Required attributes with Price Grabber

Recommended attributes with Price Grabber

How to configure my template with Nextag?

File format with Nextag

Required attributes with Nextag

Recommended attributes with Nextag

How to configure my template with Twenga?

File format with Twenga

Required attributes with Twenga

Recommended attributes with Twenga

Example of template with XML, CSV and TXT format with Twenga

How to configure my template with Idealo?

File format with Idealo

Required attributes with Idealo

Recommended attributes with Idealo

Example of template with CSV format with Idealo

How to configure my template with eBay?

File format with eBay

Required attributes with eBay

Recommended attributes with eBay

Example of template with XML format with eBay

How to use Data Feed Manager?

STEP 1: Configure Data Feed Manager in a general way

Before being able to configure your data feeds, you should configure the extension in a general way. To do so, go to

➤ SYSTEM ➤ CONFIG ➤ WYOMIND ➤ DATA FEED MANAGER.

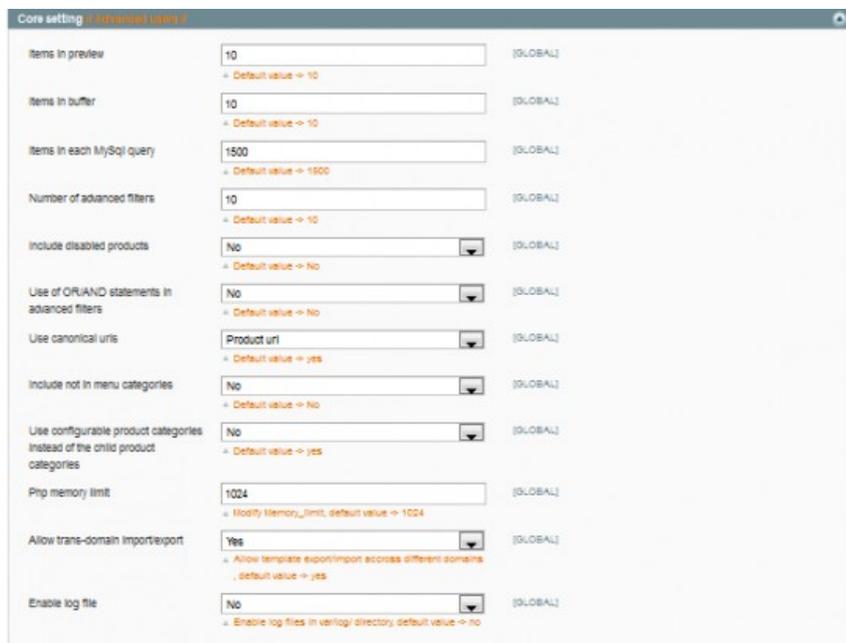
In the **Cron job reporting** tab, you can activate the cron job reporting and fill in an email address to receive the reports. You can also rename the reports as you want.



(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-cron-reports.png>)

In the **Core Settings** tab, you'll need to configure a certain number of fields.

- **Items in preview:** By default there are 10 items in preview, if you want you can reduce or increase that number.
- **Items in buffer:** By default, products are inserted into the file on the server every 10 items but you can reduce or increase that number.
- **Items in each MySQL query:** By default, there are 1500 items in each MySQL query.
- **Number of advanced filters:** By default, you have 10 advanced filters. If you need more, you can at any time increase that value.
- **Include disabled products:** you have the possibility to include or not disabled products in your data feed.
- **Use of OR/AND statements in advanced filters:** and/or statements can be used in the advanced filters simply by enabling that option.
- **Use canonical urls:** you have the choice between several options. You can choose to use the shortest category url, the longest one or the product url.
- **Include not in menu categories:** you can also include the categories of products that are not in the menu.
- **Use configurable product categories instead of the child product categories:** Use the categories of the configurable products instead of the child ones.
- **Php memory limit:** The default memory limit is 1024.
- **Allow trans-domain import/export:** Allow import and export options.
- **Enable log file:** Enable or disable log file.



(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-core-settings.png>)

STEP 2: Configure your data feeds

To create and configure your data feeds, you can go to

- CATALOG ➤ DATA FEED MANAGER ➤ DATA FEEDS.

Data Feed Manager also allows you to export and import templates into/from your Magento admin.

Export new template

To export a template, select an existing feed from

- CATALOG ➤ DATA FEED MANAGER ➤ DATA FEEDS.

From the dropdown under the **Action** column, you can choose to **Export** that data feed template. You can also choose to **Edit** the data feed and then click on **Export**.

Via both ways, a popup window opens and allows you to save the data feed template as a *.dfm* file. After that, you should find the data feed template in your **'Downloads'** folder.



(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-export-template.png>)

Import new template

Before anything, to import a data feed template, it must be a *.dfm* file.

To do so, go to

➤ CATALOG ➤ DATA FEED MANAGER ➤ DATA FEEDS

and click on **Import a template**. Browse the *.dfm* file and click on **Import now**.

Now if you go back to

➤ CATALOG ➤ DATA FEED MANAGER ➤ DATA FEEDS

you should see that the last feed of the list is the one you've just imported.



(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-import-template.png>)

Create a template

To create your own data feed, you need to go to

➤ CATALOG ➤ DATA FEED MANAGER ➤ DATA FEEDS.

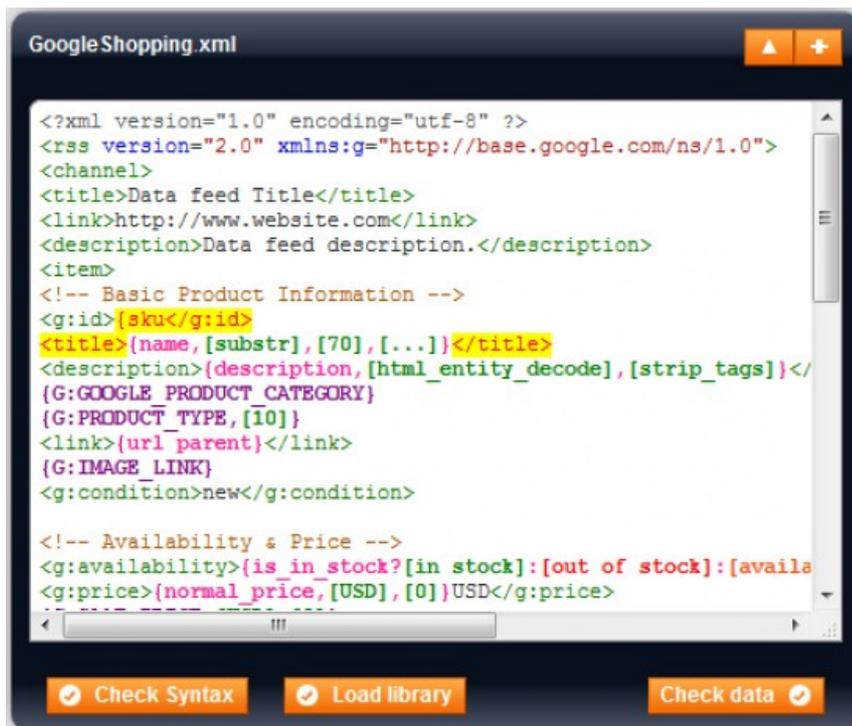
There, click on **Create new template**. You now have to configure your data feed as you want.

When creating a data feed, you should see a **blackbox** on the right of your screen. The *Data Feed Manager* extension includes that tool to help you create your feed and see what is wrong.

The blackbox is inevitably a very useful tool. This will help you to configure your data feed more quickly.

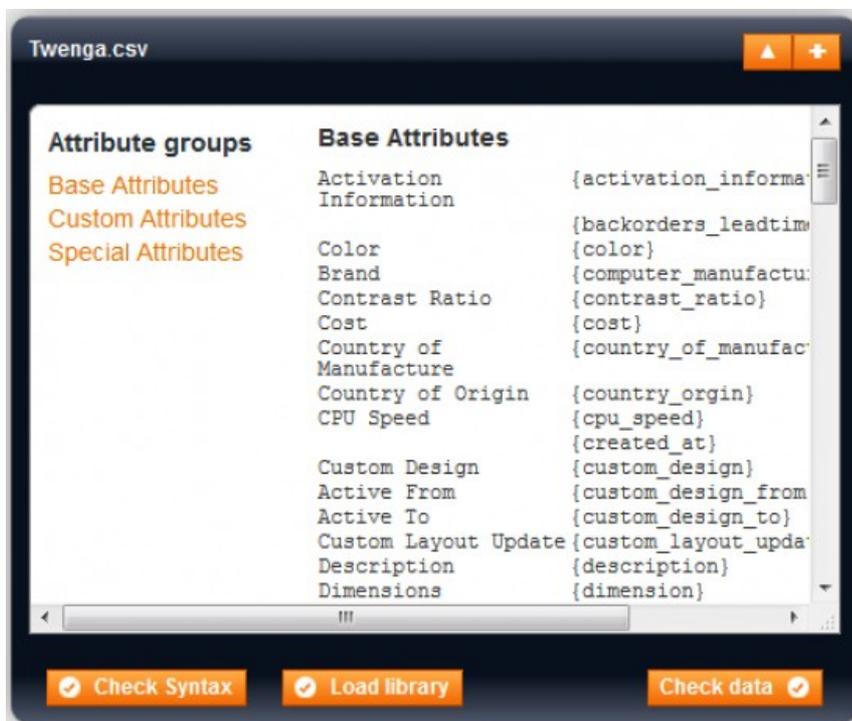
Here are the different options of the blackbox:

Check syntax: This button allows you to check if your syntax is correct. It helps you to see if you have forgotten a bracket or a coma by underlining in yellow the part of text that contains a mistake.



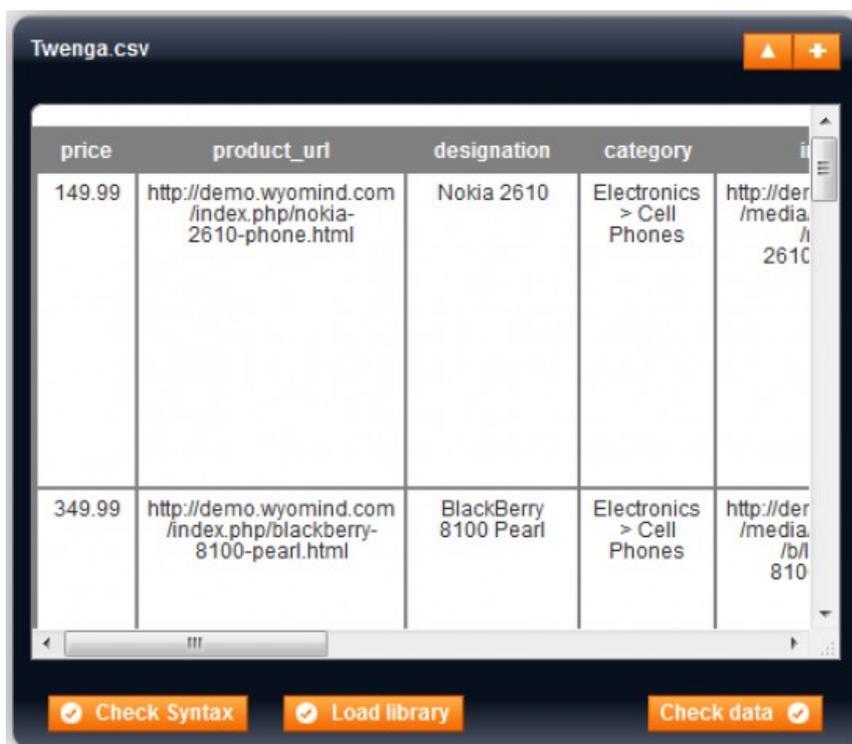
(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-check-syntax.png>)

Load Library: If you click on that button, a library of your store attributes will be displayed. This makes it easier to find very quickly which attribute you need to add into your template.



(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-load-library.png>)

Check data: When clicking on **Check Data**, you'll see a preview of your data feed. It's very useful to see if the values of your attributes are correctly retrieved or if your filters are working.



(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-check-data.png>)

Configuration

In that first tab is the main configuration of the data feed. You'll find several fields to fill in.

- **Feed name:** enter the name of your feed. For example GoogleShopping.
- **File type:** choose the format of your file between xml, csv, txt and tsv.
- **File name format:** you have several format possibilities for your file name. For example you could choose: GoogleShopping-2015-01-26.xml.
- **Encoding type:** choose the encoding type of the data feed.

- **Path:** define the path where you want the data feed to be generated. For example /feeds/.
- **File status:** enable or disable the file.
- **Store View:** define which store view is concerned.

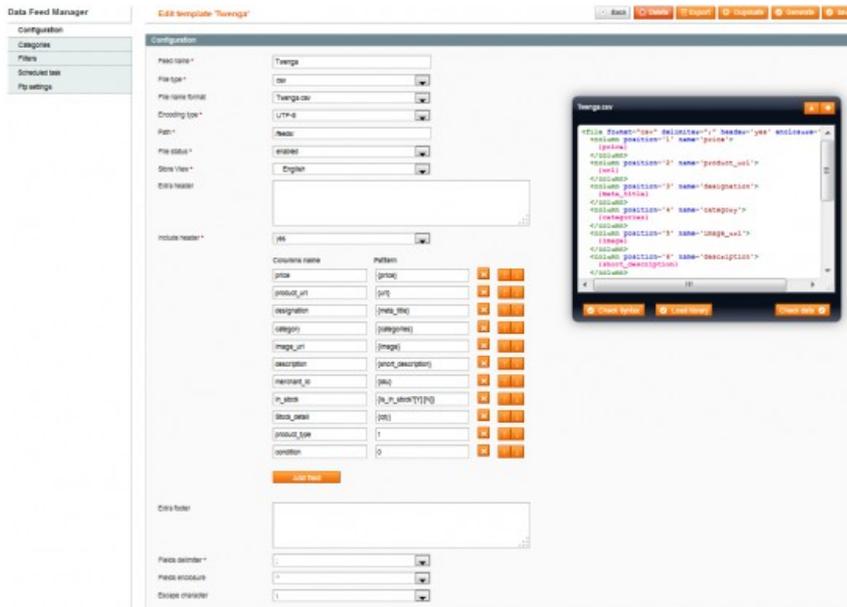
The following fields vary according to the file type you're configuring:

XML FILES

- **Enclose xml tag content inside CDATA (recommended):** enclose xml tags inside CDATA or not.
- **Remove all empty xml tags (recommended):** remove empty xml tags or not.
- **Header pattern:** Add your header pattern in that field.
- **Product pattern:** This field is inevitable to configure your product pattern.
- **Footer pattern:** Add your footer pattern in that field.

CSV, TXT, TSV FILES

- **Extra header:** It will add whatever you write on the top of your file (fit in with Amazon requirements).
- **Include header:** allows you to include or not columns names in the first row of your file.
- **Columns name:** is for the attribute name (you need to look in the shopping engine specifications to know which names are required).
- **Pattern:** works in the same way than XML files. For example, {url} will return the product's url. You can also add any static values.
- **Add field button:** click on it to add a new column to your template. The fields order can be easily reordered by using the up and down arrows.
- **Extra footer:** you can add an extra footer in that field.
- **Fields delimiter:** lets you choose which kind of delimiter you want to use to separate your fields.
- **Fields enclosure:** lets you choose which kind of enclosure you want for your data.
- **Escape character:** define the escape character between \ or ".



(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-config.png>)

Categories

In the **Categories** tab, you will find 2 subtabs that will allow you to manage your categories.

Filter the categories

You can filter your categories according to two different ways:

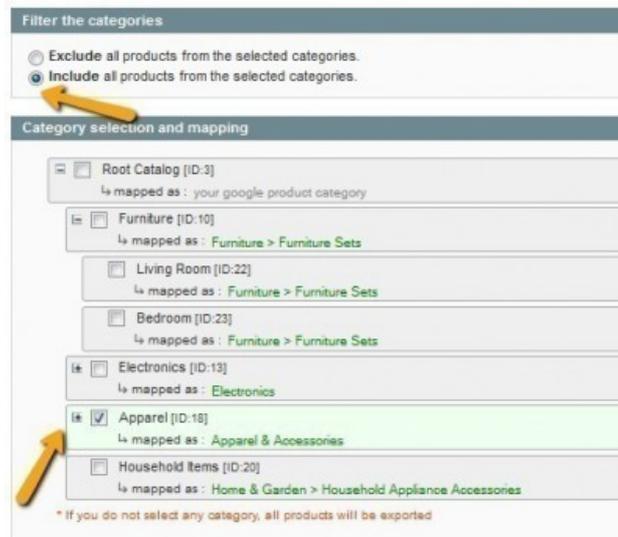
- **Exclude** all products from the selected categories.

- **Include** all products from the selected categories.

If you don't select any category, all products will be exported.

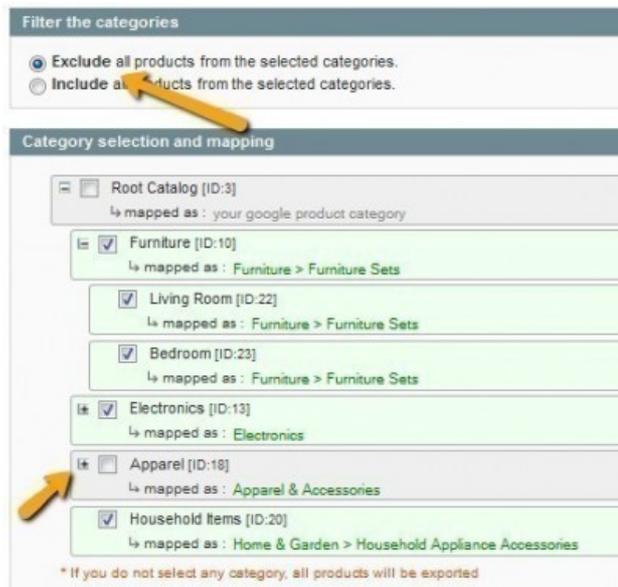
For example, if you wish to include only products of the Apparel category, you can configure your filter in two different ways.

First of all by ticking the **Include** option that will allow you to include all categories that you have selected.



(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-catego-1.jpg>)

Or via the **Exclude** option that will exclude all categories you have selected.

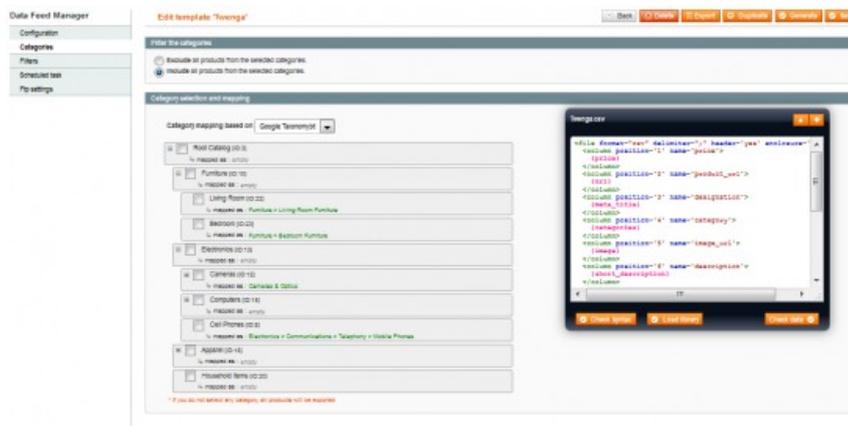


(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-catego-2.jpg>)

Category selection and mapping

In that subtab, you can map your categories to Google ones. This step is very important so that Google can make a link between their product categories and yours.

With *Data Feed Manager*, you have the possibility to change the file on which the category mapping is based. You can have a look to our [faq](#) on how to add a new taxonomy file.



(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-categ.png>)

Check our [faq](#) to know how to map your categories.

Filters

Data Feed Manager also allows you to filter your products on several levels according to their attribute sets or their visibility for example.

With *Data Feed Manager 7.3.0* and *Simple Google Shopping 9.0.0*, you can now select all the filters in one click.

Product Type

First of all, you can choose to include products to your feed according to their type. For example, only include simple, virtual and downloadable products.

Attribute Set

You have the choice to include several attribute sets. In our case, we've decided to include only attribute sets that are linked to the Apparel category.

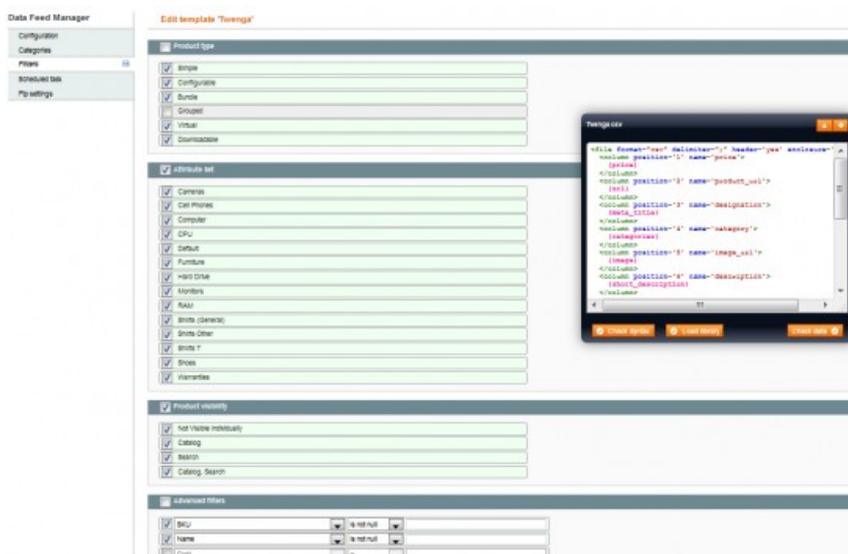
Product Visibility

Then, choose the products that will be included to the feed between the products that are not visible individually, products from the Catalog, the Search results, or both.

Advanced Filters

In the Advanced filters, you can go further by setting more personalized filters.

For example, we wanted to include only products which price is higher than 0, and which SKU and name are not null.



(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-filters.png>)

Here again, you have the possibility to configure your advanced filters according to several options.

LIKE / NOT LIKE

like: starts / ends with the character string / includes the character string. Eg:

- name like %shirt: will include all products which name ends with 'shirt'.
- name like Apple%: will include all products which name starts with 'Apple'.
- name like %women%: will include all products which name contains the 'women' character string.

not like: doesn't start / end with the character string / doesn't include the character string. Eg:

- name not like %shirt: won't include any product which name ends with 'shirt'.
- name not like Apple%: won't include any product which name starts with 'Apple'.
- name not like %women%: won't include any product which name contains the 'women' character string.



A screenshot of a search filter configuration. It shows a dropdown menu with 'Name' selected, followed by a dropdown with 'like' selected, and a text input field containing '%women%'.

(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-eg-like.png>)

IS NULL / IS NOT NULL

is null: the value is null. Eg:

- Special price is null: will include all products that don't have special prices.

is not null: the value is not null. Eg:

- SKU is not null: will include all products which SKU is not null.



A screenshot of a search filter configuration. It shows a dropdown menu with 'Special Price' selected, followed by a dropdown with 'is null' selected.

(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-eg-null.png>)

IN / NOT IN

in: is in (selection of several options). Eg:

- Brand in Acer Apple Sony: will include only products from these 3 brands.

not in: is not in (selection of several options). Eg:

- Brand not in Dell Toshiba: will include all products which brand is not Dell or Toshiba.



A screenshot of a search filter configuration. It shows a dropdown menu with 'Brand' selected, followed by a dropdown with 'in' selected, and a list of brands: Acer, Apple, Dell.

(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-eg-in.png>)

You can activate the AND/OR statement in System > Config > Wyomind > Data Feed Manager.

Scheduled Tasks

Data Feed Manager provides you a fully configurable schedule in the **Scheduled task** tab of your template configuration for each data feed. You just have to tick the days and hours when you want your feed to be generated.



A screenshot of the Data Feed Manager configuration interface. The 'Scheduled task' tab is active, showing a grid of checkboxes for scheduling the feed. The grid has columns for days of the week (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday) and rows for hours of the day (1:00 AM, 2:00 AM, 3:00 AM, 4:00 AM, 5:00 AM, 6:00 AM, 7:00 AM, 8:00 AM, 9:00 AM, 10:00 AM, 11:00 AM). The checkboxes are currently unchecked.

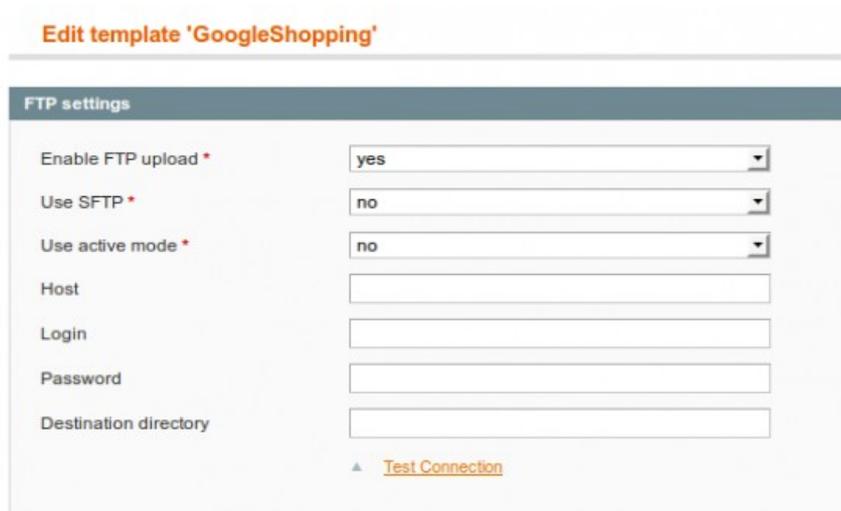
(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-scheduled-tasks.png>)

If you're facing some difficulties with scheduled tasks, or if you think your feed is not automatically generated, have a look to our [faq](#) on that subject.

FTP Settings

In *Data Feed Manager*, you can specify ftp settings to upload your data feeds.

Select a data feed and go to the **Ftp Settings** tab. There you'll be able to configure some parameters:



The screenshot shows a web interface for editing a template named 'GoogleShopping'. The main heading is 'Edit template 'GoogleShopping'' in orange. Below it is a section titled 'FTP settings' with a dark blue header. The form contains several fields: 'Enable FTP upload *' is a dropdown menu set to 'yes'; 'Use SFTP *' is a dropdown menu set to 'no'; 'Use active mode *' is a dropdown menu set to 'no'; 'Host', 'Login', 'Password', and 'Destination directory' are text input fields. At the bottom of the form is a button labeled 'Test Connection' with a small upward-pointing triangle icon.

(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-faq-ftp-yes.png>)

Enable FTP upload

Select **Yes** if you want to upload your data feed using FTP.

Use SFTP

Select **Yes** if you are using SFTP.

Secure connection (sftp) is only available in the latest version of *Data Feed Manager*.

Use Active mode

If you are using SFTP, you don't have to care about the passive/active mode.

For more info about **Active** and **Passive** mode, have a look to that documentation [🔗](#).

Host

In that field, fill in the host.

For example: ftp.wyomind.com

Login

is your login to connect to server

Password

is your password to connect to server

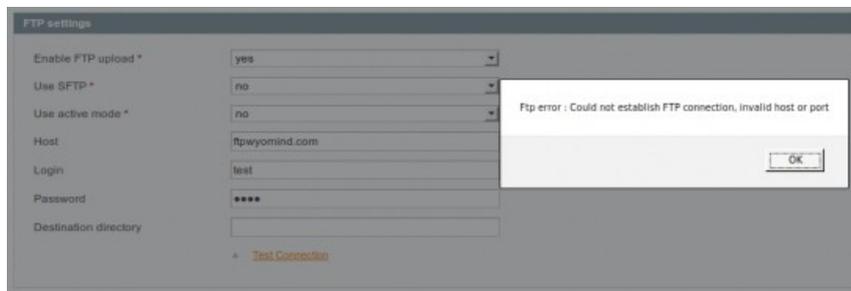
Destination directory

is a relative path from the entry directory when you connect to the ftp.

For example: "upload/" or "/".

In order to be sure your settings are correct, click on **Test Connection**.

- If you don't have an error message, it means the connection succeeded.
- If you have an error message, you'll certainly have to check your credentials.



(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-faq-ftp-dfm.png>)

The file is uploaded each time the feed is generated manually or automatically. In the case where you didn't get error message while testing the connection, you'll need to check that the file is well uploaded.

STEP 3: Generate the data feeds

Generate the data feeds manually

There are two different ways you can use to generate your data feeds:

- You can generate your data feeds manually in the grid from the action dropdown.



(<https://www.wyomind.com//media/extensions/simple-google-shopping/magento-1/userguide/simple-google-shopping-magento-generate-feeds.png>)

- Or you can do it from the configuration panel by clicking on "generate"



(<https://www.wyomind.com//media/extensions/data-feed-manager/magento-1/userguide/configuringpanel.png>)

Generate the data feeds automatically through the cron tasks

You can generate your data feeds automatically using the cron tasks.

First, check that the main cron task is enabled on your server.

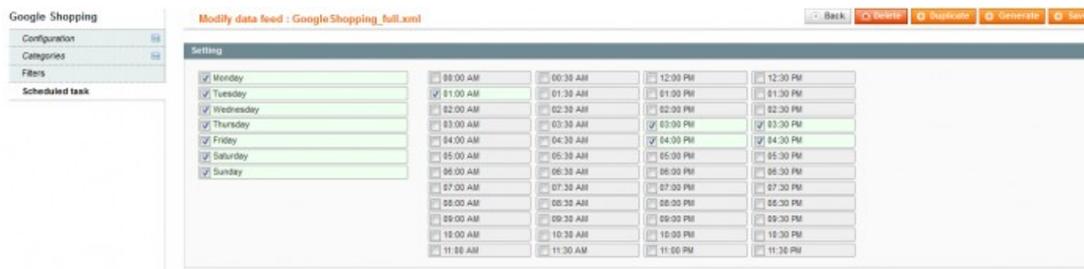


(<https://www.wyomind.com//media/extensions/simple-google-shopping/magento-1/userguide/simple-google-shopping-magento1-scheduler-is-working.png>) If it is disabled, a warning message will appear in Data Feed Manager



(https://www.wyomind.com//media/extensions/simple-google-shopping/magento-1/userguide/crontask_1.png)

You have to select the days and timeframes for the data feeds you want to update, in the *Scheduled Tasks* tab.



(<https://www.wyomind.com//media/extensions/simple-google-shopping/magento-1/userguide/simple-google-shopping-magento-scheduled-task.png>)

Manage your data feed templates thanks to Data Feed Manager

How to generate XML files with Data Feed Manager

All XML data feeds used by shopping engines, like the one used by Google Shopping, are really simple to build but often have their own specifications.

Simple Google Shopping and *Data Feed Manager* provide one or more ready-to-configure templates. These templates will allow you to build a valid and well-structured data feed quickly.

7 Things to know when creating an XML data feed with Data Feed Manager:

1. An XML data feed is using an **opening** and **closing** tag that includes a **value**.

```
<tag>value</tag>
```

2. For each product you are exporting, you need to choose which **tag** to use. Most of the time, there is a tag for **each attribute**.

```
<g:id>123456</g:id>
  <price>45.99 USD</price>
```

3. When you are building your XML template, you can include between every opening and closing tag one or several **Magento attributes**. An attribute is specified by its **code name** and must be **enclosed using brackets**.

```
<g:id>{sku}</g:id>
  <price>{price}</price>
```

4. Between every opening and closing tags, you can also include some **fixed values**. Fixed values are simply text.

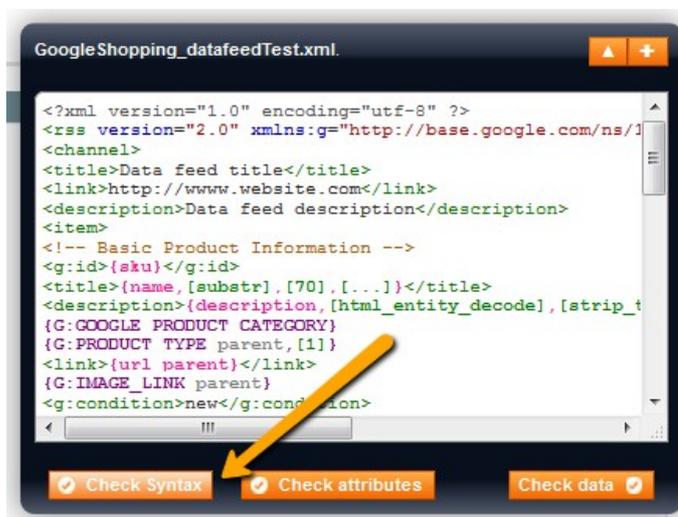
```
<g:brand>YOUR BRAND</g:brand>
```

5. You have also the possibility to add **several attributes** and **fixed values** in a same tag.

```
<title>{name},available in {color}-{size} for only ${price}</title>
```

6. You're allowed to use **PHP code** if you want to customize the pattern. You just have to write your PHP code in one single line, and it will work perfectly (check how to add PHP scripts into your templates).

7. The **syntactic coloration** in Data Feed Manager will help you to avoid any error in your template. First, the errors will be underlined in yellow. You need to correct each error before checking syntax once again. Finally, you'll be able to check data.



(<https://www.wyomind.com//media//extensions/simple-google-shopping/magento-1/userguide/simple-google-shopping-magento-check-syntax.png>)

How to generate CSV/TXT files with Data Feed Manager?

Data Feed Manager allows you to generate CSV/TXT files in addition to XML files. This particular kind of file format is required by some shopping engines like Amazon, Shopzilla, Twenga, etc.

Data Feed Manager provides you all you need to create any specific CSV/TXT file that you want.

Each shopping engine has its own requirements and column names. Some require fields enclosure, others need specific delimiters.

You may also need to add a special header to your file (for Amazon data feeds for example).

Things to know when creating a CSV/TXT data feed with Data Feed Manager

1. A CSV file consists of any number of records. Each record consists of fields, separated by some other characters or strings. There is one row per record (so, one per product).

Here is an example of CSV file content:

```
"price","product_url","designation","category","image_url","description","merchant_id","in_stock","stock_detail","product_type","condition"
"149.99","http://demo.wyomind.com/nokia-2610-phone.html","Nokia 2610","Electronics > Cell Phones","http://demo.wyomind.com/media/catalog/product"
"949.99","http://demo.wyomind.com/blackberry-9100-pearl.html","BlackBerry 9100 Pearl","Electronics > Cell Phones","http://demo.wyomind.com/media"
"299.99","http://demo.wyomind.com/sony-ericsson-w910i.html","Sony Ericsson W910i","Electronics > Cell Phones","http://demo.wyomind.com/media/cate"
"199.99","http://demo.wyomind.com/atsa-8028-pda.html","AT&T 8028 PDA Phone","Electronics > Cell Phones","http://demo.wyomind.com/media/catalog"
"150.00","http://demo.wyomind.com/samsung-me-a900e-sa.html","Samsung ME-A900M Ase Phone","Electronics > Cell Phones","http://demo.wyomind.com/MS"
"2299.99","http://demo.wyomind.com/apple-macbook-pro-ma4641l-a-15-4-notebook-pc-2-0-ghz-intel-core-duo-1-gb-ram-100-gb-hard-drive-superdrive.html"
"1799.99","http://demo.wyomind.com/acer-ferzari-3200-notebook-computer-pc.html","Acer Ferzari 3200 Notebook Computer PC","Electronics > Computer"
"2699.99","http://demo.wyomind.com/sony-viao-vgn-tan27n-b-11-1-notebook-pc.html","Sony VIAO VGN-TXN27N/S 11.1" Notebook PC (Intel Core Solo E60"
"1599.99","http://demo.wyomind.com/toshiba-ate111ce-a135-a407-155-4-notebook-pc-intel-pentium-dual-core-processor-t2080-1-gb-ram-120-gb-hard-di"
"15.99","http://demo.wyomind.com/cn-clogs-beach-garden-clog-3.html","CN Clogs Beach/Garden Clog","http://demo.wyomind.com/media/catalog/produ"
"134.99","http://demo.wyomind.com/asico-men-s-pel-kajano-ii-8.html","ASICM Men's PEL-Kajano® XII","Apparel > Shoes > Mens","http://demo.wyomind"
"69.99","http://demo.wyomind.com/steven-by-steve-madden-pryme-pump-4.html","Steven by Steve Madden Pryme Pump","Apparel > Shoes > Womens","http:"
"89.99","http://demo.wyomind.com/nine-west-women-s-lucero-pump-3.html","Nine West Women's Lucero Pump","Apparel > Shoes > Womens","http://demo.w"
"159.99","http://demo.wyomind.com/ecco-womens-golf-fleaser-golf-shoe-3.html","ECCO Womens Golf Fleaser Golf Shoe","Apparel > Shoes > Womens","http"
"160.99","http://demo.wyomind.com/kenneth-cole-new-york-men-s-con-venge-slip-on-9.html","Kenneth Cole New York Men's Con-venge Slip-on","Apparel"
"18.00","http://demo.wyomind.com/coleseco-functioning-on-impatience-t-shirt-small.html","Coleseco: Functioning On Impatience T-Shirt","Apparel >"
"12.00","http://demo.wyomind.com/link-eater-kyllion-bombear-destroyed-tee-sm.html","Link Eater: Kyllion Bombear Destroyed Tee","Apparel > Shirts","h"
"15.00","http://demo.wyomind.com/the-only-children-paisley-t-shirt-sm.html","The Only Children: Paisley T-Shirt","Apparel > Shirts","http://demo"
"13.50","http://demo.wyomind.com/solof-the-rock-wad-roll-destroyer-lol-cat-t-shirt-sm.html","Solof The Rock And Roll Destroyer: LOL Cat T-shirt"
```

(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-csv-file-example.jpg>)

2. You need to define **delimiters** to separate your fields (; , ' , or tab for example) and **enclosures** for your data (" or ').
3. You can enter the attribute names in each **Columns name** fields (check the shopping engine specifications to know which names are required). For example:

(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-csv-file-type2.jpg>)

4. You can include in each **Pattern** field one or several Magento attributes. An attribute is specified by its **code name** and must be **enclosed using brackets**. For example, to return the product's sku, you should use:

{sku}

5. You can also include some **fixed values**. Fixed values are simply text. For example, instead of the {manufacturer} attribute, you could use directly:

YOUR BRAND

6. You have also the possibility to add **several attributes** and **fixed values** in a same field. For example, instead of {url}, you could use:

{url}?_campaign=google&_adwords=test123

7. You're allowed to use **PHP code** if you want to customize the pattern. You just have to write your PHP code **in one single line**, and it will work perfectly (check how to add PHP scripts into your templates).

8. You can add a field by using the button **Add field**. The fields order can be easily reordered by using the **up and down arrows**.

9. The **syntactic coloration** in *Data Feed Manager* will help you to avoid any error in your template. First, the errors will be **underlined in yellow**. You need to correct each error before **checking syntax** once again. Finally, you'll be able to **Check data**.

price	product_url	description	category	image_url	description	merchant_id	in_stock	stock_amount	product_type	condition
149.99	http://demo.wyomind.com/nokia-2610-phones.html	Nokia 2610	Electronics > Cell Phones	http://demo.wyomind.com/media/catalog/product/nob/nokia-2610-2.jpg	The words "entry level" no longer mean "low-end," especially when it comes to the Nokia 2610. Offering advanced media and calling features without breaking the bank.	n2610	Y	163	1	0
349.99	http://demo.wyomind.com/blackberry-8100-pearl.html	BlackBerry 8100 Pearl	Electronics > Cell Phones	http://demo.wyomind.com/media/catalog/product/abb/blackberry-8100-pearl-2.jpg	The BlackBerry 8100 Pearl is a departure from the form factor of previous BlackBerry devices. This BlackBerry handset is far more phone-like, and RIM's engineers have managed to fit a QWERTY keyboard onto the handset's slim frame.	bb8100	Y	787	1	0
399.99	http://demo.wyomind.com/sony-ericsson-w610a.html	Sony Ericsson W610	Electronics > Cell Phones	http://demo.wyomind.com/media/catalog/product/sony-ericsson-w610a-2.jpg	The W610 follows a long tradition of beautifully designed and crafted phones from Sony Ericsson. The same candy-bar style that graced the W600 is here.	sw610a	Y	8	1	0

(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-csv-preview.jpg>)

Attribute management in the Data Feed Manager templates

All the default Magento attributes and also all your own attributes can be retrieved in any template. For example :

```
<product>{name}</product>
```

Some Magento attributes like price, image or url may have further configuration options that will correspond to your specific data feed needs. For example:

```
<price>{price,[EUR],[FR]}</price>
```

Some Magic attributes are unique to the Google Shopping data feeds. These attributes act as shortcuts to create complete portions of XML in your data feed. For example:

```
{G:PRODUCT_TYPE}
```

You can use the OR operand with all these attributes (without option). See example below:

```
{attributeA | attributeB}
```

If attributeA exists then the value will be attributeA, otherwise, the value will be attributeB



(<https://www.wyomind.com//media//extensions/simple-google-shopping/magento-1/userguide/simple-google-shopping-magento-operand.png>)

Attribute specifications

PARENT ATTRIBUTES

Most of the time simple products are related to parent products like configurable, grouped and bundle products. That is why you might need to retrieve Parent attribute values instead of simple product attribute values such as url, image, description...

All the above attributes (Basic, Special and Magic) may be used to retrieve a Parent attribute value instead of a product attribute value.

All options are still available, eg: {description parent,[htmlentities]}

```
{attribute_code parent}
```

Retrieves the first parent attribute value.

The parent value is retrieved **only if the simple (item) product is associated with a parent product**, if not, its own attribute value is retrieved.

The parent product may be a configurable, a grouped or a bundle product. The first parent product that matches the simple product is used to get the attribute value.

Eg:

```
{G:PRODUCT_TYPE parent}
  <g:product_type>{categories parent,[last]}</g:product_type>
```

```
{attribute_code configurable}
```

Retrieves the first configurable parent attribute value.

The configurable product value is retrieved **only if the simple product is associated to a configurable product**, if not, its own attribute value is retrieved.

Eg:

```
{G:IMAGE_LINK configurable}
  <g:image_link>{image configurable}</g:image_link>
```

{attribute_code grouped}

Retrieves the first grouped parent attribute value.

The grouped product value is retrieved **only if the simple product is associated to a grouped product**, if not, its own attribute value is retrieved.

Eg :

```
{G:GOOGLE_PRODUCT_CATEGORY grouped}
  <link>{url grouped}</link>
```

{attribute_code bundle}

Retrieves the first bundle parent attribute value.

The bundle product value is retrieved **only if the simple product is associated to a bundle product**, if not, its own attribute value is retrieved.

Eg:

```
{G:SALE_PRICE bundle}
  <g:price>{price bundle}</g:price>
```

PRODUCT IDENTIFIERS

{id}

Retrieves the product id.

For example:

```
<g:id>{id}</g:id>
```

{G:ID}

Is the shortcut for:

```
<g:id>{sku}</g:id>
```

You can retrieve the id instead of the sku by adding an option:

```
{G:ID, [id]}
```

{G:ITEM_GROUP_ID}

Is the shortcut for:

```
<g:item_group_id>{sku configurable}</g:item_group_id>
```

{sku}

Retrieves the product SKU

For example:

```
<g:mpn>{sku}</g:mpn>
```

{brand}

Retrieves the brand of the product.

For example:

```
<g:brand>{brand}</g:brand>
```

{manufacturer}

Retrieves the brand of the product.

For example:

```
<g:manufacturer>{manufacturer}</g:manufacturer>
```

BASIC INFORMATION

{inc}

Retrieves an auto-incremented value, starting at 1.

For example:

```
<g:increment>{inc}</g:increment>
```

{name}

Retrieves the product name.

Par exemple:

```
<g:title>{name}</g:title>
```

{G:TITLE}

Is the shortcut for:

```
<title>{name}</title>
```

{description}

Retrieves the description of the product.

For example:

```
<g:description>{description}</g:description>
```

{G:DESCRIPTION}

Is the shortcut for:

```
<description>{description}</description>
```

You can retrieve the short description instead by adding a parameter:

```
{G:DESCRIPTION, [short]}
```

{short_description}

Retrieves the short description of the product.

For example:

```
<g:description>{short_description}</g:description>
```

{condition}

Retrieves the condition of the product.

For example:

```
<g:condition>{condition}</g:condition>
```

{weight}

Retrieves the weight of the product.

For example:

```
<g:weight>{weight}</g:weight>
```

PRICES AND PROMOTIONS

{price}

Retrieves the normal price or the special price if it exists.

If prices are exclusive of VAT and the product is liable to a unique VAT rate, the price retrieved will include this rate. Otherwise, the price exclusive of tax is retrieved.

For example:

```
<g:price>{price}</g:price>
```

You also have the possibility to use some parameters with this attribute:

- **currency** is for currency conversion. This option automatically converts the original product price to a specified currency. It uses an exchange rate already stored by you in your admin panel. By default, your local currency is used.
- **vat_rate** is a floating value between 0.00 and 100.00 that represents an arbitrary VAT rate to apply. This option rewrites the VAT rule for the product except if the product doesn't have any tax class.

```
<g:price>{price,[currency],[vat_rate]}</g:price>
```

Here are some examples:

To convert the prices to EUR and apply a 20% VAT rate:

```
<g:price>{price,[EUR],[20.0]}</g:price>
```

To convert the prices to USD and not apply any VAT rate:

```
<g:price>{price,[USD],[0]}</g:price>
```

If the prices of your catalog are already including VAT:

```
<g:price>{price,[EUR]}</g:price>
```

To use an ISO2 code corresponding to one of the countries for which you specified a VAT rate in your tax settings (from Simple Google Shopping >= 5.1.0, and Data Feed Manager >= 3.0.0):

```
<g:price>{price,[EUR],[FR]}</g:price>
```

This code may also be completed by an area code:

```
<g:price>{price,[USD],[US/NY]}</g:price>
```

To apply the VAT corresponding to a country/area:

```
<g:price>{price,[USD],[-US/NY]}</g:price>
```

To retrieve prices including different price values for simple products associated to a configurable one (from Data Feed Manager v7.3):

```
<g:price>{price parent,[EUR],[20.0]}</g:price>
```

{G:PRICE}

Is the shortcut for:

```
<g:price>{price}</g:price>
```

{normal_price}

Retrieves the price and ignores any existing special price value.

For example:

```
<g:price>{normal_price}</g:price>
```

You also have the possibility to use the same parameters than the {price} attribute.

{special_price}

Retrieves the promotional price (special price) if exists.

If this price doesn't exist the value is null.

For example:

```
<g:price>{special_price}</g:price>
```

You also have the possibility to use the same parameters than the {price} attribute.

{price_rules}

Retrieves either the price or the special price or the price including the catalog price rules if indicated.

For example:

```
<g:price>{price_rules}</g:price>
```

You also have the possibility to use some parameters with this attribute:

- **currency** is for currency conversion. This option automatically converts the original product price to a specified currency. It uses an exchange rate already stored by you in your admin panel. By default, your local currency is used.
- **vat_rate** is a floating value between 0.00 and 100.00 that represents an arbitrary VAT rate to apply. This option rewrites the VAT rule for the product except if the product doesn't have any tax class.
- **customer_group_id** allows you to determine which price rules must apply for a given custom group id (by default the "NOT LOGGED IN" customer group applies)

```
<g:price>{price_rules,[currency],[vat_rate],[customer_group_id]}</g:price>
```

{group_price}

Retrieves the group price if it exists. Otherwise, it displays the normal price.

For example:

```
<g:price>{group_price,[1],[USD],[0]}</g:price>
```

You also have the possibility to use some parameters with this attribute:

- **customer_group** is the id of the customer group for which the group price is valid (check in  CUSTOMERS  CUSTOMER GROUPS).
- **currency** is for currency conversion. This option automatically converts the original product price to a specified currency. It uses an exchange rate already stored by you in your admin panel. By default, your local currency is used.
- **vat_rate** is a floating value between 0.00 and 100.00 that represents an arbitrary VAT rate to apply. This option rewrites the VAT rule for the product except if the product doesn't have any tax class.

```
<g:price>{group_price,[customer_group],[currency],[vat_rate]}</g:price>
```

{G:SALE_PRICE}

Is the shortcut for:

```
<g:sale_price>{special_price}</g:sale_price>
    <g:sale_price_effective_date>{special_from_date}/{special_to_date}</g:sale_price_effective_date>
```

Must be used in combination with {normal_price}:

```
<g:price>{normal_price}</g:price>
    {G:SALE_PRICE}
```

Options: see [{price}](#)

{is_special_price}

Retrieves either 1 if a special price exists or 0 if it doesn't.

{is_special_price?[value_if_yes]:[value_if_no]}

- **value_if_yes** value to retrieve if a special price exists. By default 1.
- **value_if_no** value to retrieve if no special price exists. By default 0.

For example:

```
<g:is_special_price>{is_special_price?[yes]:[no]}</g:is_special_price>
```

{tier_price}

Retrieves the tier price if it exists, otherwise, the value is null.

For example:

```
<g:price>{tier_price}</g:price>
```

You also have the possibility to use some parameters with this attribute:

- **customer_group** is the id of the customer group for which the tier price is valid (check in [CUSTOMERS > CUSTOMER GROUPS](#)).
- **index** is the index of the Tier Price you want to retrieve.
- **currency** is for currency conversion. This option automatically converts the original product price to a specified currency. It uses an exchange rate already stored by you in your admin panel. By default, your local currency is used.
- **vat_rate** is a floating value between 0.00 and 100.00 that represents an arbitrary VAT rate to apply. This option rewrites the VAT rule for the product except if the product doesn't have any tax class.

```
<g:price>{tier_price,[customer_group],[index],[currency],[vat_rate]}</g:price>
```

{min_price}

Retrieves the minimum price.

For example:

```
<g:price>{min_price,[EUR],[20.0]}</g:price>
```

{min_price} must be used for Bundle and Grouped products.

{max_price}

Retrieves the maximum price.

For example:

```
<g:price>{max_price,[EUR],[20.0]}</g:price>
```

{max_price} must be used for Bundle and Grouped products.

{G:PROMOTION_ID}

Retrieves the product promotion id if exists.

To know more about the use of this attribute, check out our doc on Google Merchant Promotions [↗](#) extension.

LINKS

{url}

Retrieves the product's absolute url.

For example:

```
<g:link>{url}</g:link>
```

The url attribute can receive the currency code as an url parameter to display the landing page with a specific currency, for example:

```
<g:link>{url}?currency=USD</g:link>
```

{G:LINK}

Is the shortcut for:

```
<link>{url}</link>
```

{image}

Retrieves the product's main image.

{image,[index]}

- **index** is the index of any additional images to retrieve. By default 0 (the main image).

For example:

```
<g:image_link>{image,[0]}</g:image_link>
```

{G:IMAGE_LINK}

Is the shortcut for:

```
<g:image_link>{image}</g:image_link>  
  <g:additional_image_link>{image,[1]}</g:additional_image_link>  
  <g:additional_image_link>{image,[2]}</g:additional_image_link>  
  ...
```

{host}

Retrieves the host url, for example *https://www.website.com*.

An example with attribute {small_image}:

```
<g:image_link>{host}/media/catalog/{small_image}</g:image_link>
```

{url_key}

Retrieves the product canonical relative url.

For example:

```
<g:url>https://www.mywebsite.com/mystore_code/{url_key}</g:url>
```

CATEGORIES

{categories}

Retrieves a list of category paths separated by "," in which the product may be found

{categories,[number_of_category_path],[from_category_level],[number_of_categories_in_each_path]}

- **number_of_category_path**, by default infinity
- **from_category_level**, by default 1 (root catalog)
- **number_of_categories_in_each_path**, by default infinity

Example:

```
<g:categories>{categories,[INF],[1],[INF]}</g:categories>
```

2 arguments can be used to specify the path separator and the category separator:

{categories,[path_separator],[category_separator]}

- **path_separator**, by default "coma" (,).
- **category_separator**, by default "higher than" (>)

Example:

```
<g:categories>{categories,[,],[>]}</g:categories>
```

Only one argument can be used to get the first or the last (deepest) category path:

Example:

```
<g:categories>{categories,[last]}</g:categories>  
<g:categories>{categories,[first]}</g:categories>
```

A complete list of all categories per product can be retrieved with the following method:

```
<? $product->getCategoriesArray(); ?>
```

{G:PRODUCT_TYPE}

Has no equivalent, this attribute creates a list of all the categories where the product may be found:

```
<g:product_type>category A > sub-category B </g:product_type>  
<g:product_type>category A > sub-category B > sub-category C</g:product_type>
```

This attribute generates a tag for each category in which the product is classified.

Options: see **{categories}** [↗](#)

{category_mapping}

Retrieves any new category that you have assigned (in the Categories filter) to the item's own product category. If no mapping is assigned to the first category found, the second category will be used etc.

{category_mapping,[index]}

- **index:** index of the category mapping

For example:

```
<category>{category_mapping,[0]}</category>
  <category>{category_mapping,[1]}</category>
  <category>{category_mapping,[2]}</category>
  <category>{category_mapping,[3]}</category>
  <category>{category_mapping,[4]}</category>
```

Will retrieve a maximum of 5 category mappings in 5 separate tags.

{G:GOOGLE_PRODUCT_CATEGORY}

Is the shortcut for:

```
<g:google_product_category>{category_mapping,[0]}</g:google_product_category>
```

This magic attribute retrieves the Google category that you assigned (in filter section/category filter) of the first category found to which the product is linked. If no Google category is assigned to the first category found, the second category will be used, if no Google category is assigned to the second, the third will be used etc.

To find the Google categories that match with your categories:

<https://www.google.com/support/merchants/bin/answer.py?answer=160081> ↗

Options: see {category_mapping} ↗

{attribute_set}

Retrieves the attribute set in which the product is.

For example:

```
<attribute_set>{attribute_set}</attribute_set>
```

That attribute can be useful for your Adwords campaigns.

AVAILABILITY

{status}

Retrieves either "enabled" if the product is enabled or "disabled" if the product is disabled.

{status,[value_if_enabled],[value_if_disabled]}

You can modify the default values:

- **value_if_enabled:** value to retrieve if the product is enabled
- **value_if_disabled:** value to retrieve if the product is disabled

For example:

```
<g:status>{status,[enabled],[disabled]}</g:status>
```

{stock_status}

Retrieves the stock status "in stock" or "out of stock".

For example:

```
<g:availability>{stock_status,[instock],[outofstock]}</g:availability>
```

{G:AVAILABILITY}

is the shortcut for:

```
<g:availability>{is_in_stock?[in stock]:[out of stock]:[available for order]}</g:availability>
```

Options: see [{is_in_stock}](#)

{is_in_stock}

Retrieves either "in stock" if the product is in stock, "out of stock" if the product is out of stock or "available for order" if the product is backordered.

{is_in_stock?[value_if_in_stock]:[value_if_out_of_stock]:[value_if_preorder]}

- **value_if_in_stock** value to retrieve if the product is in stock
- **value_if_out_of_stock** value to retrieve if the product is out of stock
- **value_if_preorder** value to retrieve if the product is backorderable

For example:

```
<g:availability>{is_in_stock?[in stock]:[out of stock]:[preorder]}</g:availability>
```

{qty}

Retrieves the available quantity.

The quantity retrieved for the configurable products is based on the sum of simple products.

For example:

```
<g:availability>{qty}</g:availability>
```

{min_sale_qty}

Retrieves the minimal sale quantity.

For example:

```
<g:min_sale_qty>{min_sale_qty}</g:min_sale_qty>
```

REVIEWS

{review_count}

Retrieves the total count of all the reviews that your customers have left for the product.

{review_count,[storeview_id]}

- **storeview_id** is the id of your store view. If not specified the current store view is used.
"*" will count all reviews for all store views.

For example:

```
<g:product_review_count>{review_count,[*]}</g:product_review_count>
```

{review_average}

Retrieves the average score of all the evaluations that your customers have left for the product.

{review_average,[storeview_id],[base_value]}

- **storeview_id** id of your store view. If not specified the current store view is used.
"*" will count all reviews for all store views.
- **base_value** the base number. By default 5.

For example:

```
<g:product_review_average>{review_average,[*],[3]}</g:product_review_average>
```

{G:PRODUCT_REVIEW}

Is the shortcut for :

```
<g:product_review_average>{review_average}</g:product_review_average>  
  <g:product_review_count>{review_count}</g:product_review_count>
```

Options: see [{review_average}](#)

PRODUCT VARIANTS

{gender}

Retrieves the gender of the product.

For example:

```
<g:gender>{gender}</g:gender>
```

{age_group}

Retrieves the age group of the product.

For example:

```
<g:age_group>{age_group}</g:age_group>
```

{color}

Retrieves the color of the product.

For example:

```
<g:color>{color}</g:color>
```

{size}

Retrieves the size of the product.

For example:

```
<g:size>{size}</g:size>
```

{material}

Retrieves the material of the product.

For example:

```
<g:material>{material}</g:material>
```

{pattern}

Retrieves the pattern of the product.

For example:

```
<g:pattern>{pattern}</g:pattern>
```

BASIC OPTIONS

Basic attributes are the default attributes defined in Magento and also include any attributes you have defined for your products. (eg: name, description, short_description, weight, size, color)

You can apply one or more options to all of these attributes by using the following syntax:

```
{attribute_code,[option1],[option2],[option3],[...]}
```

All options are executed in the written order eg:

```
<title>{name}</title>  
    <title>{name,[htmlentities],[strtolower]}</title>  
    <title>{name,[htmlentities],[strtolower],[ucwords]}</title>
```

will produce:

```
<title>NOKIA@ 2610 Phone</title>  
    <title>nokia&copy; 2610 phone</title>  
    <title>Nokia&copy; 2610 Phone</title>
```

{attribute_code,[strip_tags]}

Remove all html tags.

For example:

```
<description>{description,[strip_tags]}</description>
```

{attribute_code,[cleaner]}

Remove all characters that don't belong to the utf8 charset.

For example:

```
<title>{name,[cleaner]}</title>
```

{attribute_code,[substr],[max_length],[end_string]}

Truncates the value.

- **max_length**: maximum length of the value
- **end_string**: string that will be added at the end of the string. By default: [...]

For example:

```
<description>{description,[substr],[50],[...]}</description>
```

{attribute_code,[strtoupper]}

Transforms the value to upper case.

For example:

```
<title>{name,[strtoupper]}</title>
```

{attribute_code,[strtolower]}

Transforms the value to lower case.

For example:

```
<title>{name,[strtolower]}</title>
```

{attribute_code,[inline]}

Removes all EOL characters from the value.

For example:

```
<description>{description,[inline]}</description>
```

{attribute_code,[html_entity_decode]}

Converts all HTML entities to their applicable characters.

For example:

```
<description>{description,[html_entity_decode]}</description>
```

{attribute_code,[htmlentities]}

Converts all applicable characters to HTML entities.

For example:

```
<description>{description,[htmlentities]}</description>
```

{attribute_code,[implode],[separating_string]}

Joins all array elements in a string.

- **separating_string**: string used between each value. By default it is a coma (,).

May be useful for multi-select attributes like *color*.

For example:

```
<g:color>{color,[implode],[,]}</g:color>
```

will give: *black, blue, red, green*.

{attribute_code,[float],[decimals]}

Converts a numeric value to a floating number.

- **decimals**: number of floating number. By default [0].

For example:

```
<g:shipping_weight>{weight,[float],[2]}kg</g:shipping_weight>
```

STOCK IN THE CHANNEL USERS

The Stocks in The Channel users can use the following attributes in order to build their data feeds :

{SC:URL}

Retrieves the canonical url of the product :

```
<link>https://www.example.com/product.html</link>
```

Can be used in Data Feed Manager with {sc:url}

{SC:EAN}

Retrieves an unique ean value of the product:

```
<g:ean>1234567891234</g:ean>
```

Can be used in Data Feed Manager with {sc:ean}

{SC:IMAGES}

Retrieves the associated image of each product :

```
<g:image_link>https://www.example.com/media/catalog/image.jpg</g:image_link>
```

Can be used in Data Feed Manager with : {sc:images}

{SC:DESCRIPTION}

Retrieves the description for the product:

```
<g:description>...product description ...</g:description>
```

A default value is retrieved when the description is empty.

Can be used in Data Feed Manager with {sc:description}

PHP API for Data Feed Manager

With Data Feed Manager you can use PHP scripts in different ways to fully customize and control the data feed output.

■ Inline PHP scripts

Scripts that are added directly to the data feed template surrounded by PHP short tags (<? ... ?>)

For Data Feed Manager only

■ Custom attributes

Scripts that can be called by using the custom attribute name just like you do for any attribute in your data feed template

■ Custom functions

Functions that can be called by using the function name and additional parameters and that can be applied to any attribute in your data feed template

API coding standards

The Data Feed Manager PHP API uses the exact same syntax and standards than any PHP coding.

Attribute calls

For performance reasons, Data Feed Manager only loads in the collection of products the attributes that are explicitly required in the data feed template.

This means that **only the attributes that are called using the template placeholders, {attribute_code}, will be added to the collection of products.**

As a consequence, when you try to access the value of an attribute from the *\$product* object, **you must be sure that the attribute has been added to the collection.**

Most of the time when the returned value is null for a given attribute, it means the attribute is not loaded in the collection of products.

The best way to be sure that all the attributes you need are well loaded in the collection, is to **add at the beginning of your template a simple script that lists all the attributes required.**

```
<?
    /* list of required attributes
    * {attribute_1} {attribute_2} {attribute_3} {attribute_4}...
    */
    ?>
```

Then you should be able to get the values of these attributes in your PHP script:

```
<? return $product->getAttribute1();?>
    <? return $product->getAttribute2();?>
    <? return $product->getAttribute3();?>
    ....
```

PHP functions in attribute calls

You can use any PHP function to apply in the attribute calls even if the function requires one or more additional arguments, for instance:

```
{name, [str_replace("Apple", "Banana", $self)]}
```

You can use several consecutive functions:

```
{ean,[str_pad($self,10,"0",STR_PAD_LEFT)],[str_replace("X","0",$self)]}
```

Conditional values

To output values based on different conditions, use PHP conditional statements:

■ if/else

```
<?
    /* attribute {sku} required */
    if ($product->getSku()=="abc"){
        return "Alphabetic";
    }
    elseif($product->getSku()=="123"){
        return "Numeric";
    }
    else{
        return "Mixed";
    }
?>
```

■ switch/case

```
<?
    /* attribute {sku} required */
    switch($product->getSku()){
        case "abc":
            return "Alphabetic";
            break;
        case "123":
            return "Numeric";
            break;
        default:
            return "Mixed";
    }
?>
```

Opening and closing tags

When using the PHP API inline directly in your data feed template, you must use the opening and closing short PHP tags (<? ... ?>)

```
<?
    if($product->getData('size')=="X-Large") {
        return "XL";
    }
?>
```

Outputting a value

■ Return

Any value that you want to output in your data feed, must be returned with the return instruction.

```
<? return "Don't miss! ".$product->getTitle(); ?>
    /* This script adds for each product in data feed a new output line
    * Don't miss! Iphone 5
    * Don't miss! Galaxy note 7
    * ...
    */
```

■ **echo, var_dump, print_r**

Any value printed in the data feed through these functions will be rendered at the top of the data feed and will not be added to the final file.

Such functions must be used for debugging purposes.

```
<? var_dump($product->getTitle()); ?>
    /* This script adds for each product in data feed a new debugging trace
    * String(8) "Iphone 5"
    * String(13) "Galaxy note 7"
    * ...
    */
```

Variables scope

All variables available in PHP scripts have a local scope, this means they can't be shared between several scripts.

If you wish to share a variable with different scripts all along the data feed generation process, then the variable must be set as a **global variable**.

```
<?
    global $increment;
    $increment++;
    return $increment;
    /* This script will output a incremental $variable
    * 1
    * 2
    * 3
    * ...
    */
?>
```

API Objects, Variables and Methods

Object \$product

\$product is an instance of *Mage_Catalog_Model_Product*. It contains all methods available by default and some other specific to Data Feed Manager.

The list below is not exhaustive but lists the most useful methods.

- **\$product->getCategoriesArray()**
Retrieves an array of arrays that describes the categories to which belongs the products
- **\$product->getCategoryIds()**
Lists all category ids in which the product may be found
- **\$product->getVisibility()**
Returns the visibility of the product (1 = not visible individually, 2 = catalog, 3 = search, 4 = catalog and search)
- **\$product->getStatus()**
Returns the status of the product (1 = enabled, 2 = disabled)
- **\$product->getData(\$attributeCode)**
Returns the value of the given attribute code
- **\$product->getAttributeText(\$attributeCode)**
Returns the textual value of the given attribute code

Object \$this

\$this is an instance of *Wyomind_datafeedmanager_Model_Feed*. It provides a few useful methods.

- **\$this->hasParent(\$product, \$type="parent")**

Returns a Boolean value whether a product has a parent from the given type (parent, configurable, grouped, bundle) or not

- **\$this->getParent(\$product, \$type="parent", \$strict=false)**

Returns a product object of the given parent type (parent, configurable, grouped, bundle) if available or the product object passed as 1st parameter when \$strict parameter is set on false.

- **\$this->skip()**

Skips the current product, useful to filter out some products based on specific condition

String | Integer | Boolean | Array \$self

\$self is the value of the attribute for the current product.

This variable is available only in the custom attributes, and in the custom functions.

Variable \$param

The variable **\$param** contains an array of the parameters that have been passed to the custom function.

```
{{ean,[str_pad],[10],[0]}}
```

The custom function *str_pad*, should be written as follows

```
<?
    /* var_dump($param) gives
    * Array(3)[
    *   0 => String(7) "str_pad",
    *   1 => String(2) "10"
    *   2 => String(1) "0"
    * ]
    */
    return str_pad($self,$param[1],$param[2],STR_PAD_LEFT);
?>
```

This variable is available only for the custom functions.

Custom attributes use

Data Feed Manager allows you to create your own attributes and to add them to your data feed.

To create custom attributes from your Magento admin, go to:

🔍 CATALOG 🔍 DATA FEED MANAGER 🔍 CUSTOM ATTRIBUTES

Some templates of use are loaded by default when installing the extension.

Here is how to proceed:

1. Create a new attribute.



(<https://www.wyomind.com//media/extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-custom-att-create.png>)

2. Fill in the attribute code with the name of your custom attribute

(<https://www.wyomind.com//media/extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-custom-att.png>)

3. Add your custom script. No need to add the PHP tags (<? ... ?>)

(<https://www.wyomind.com//media/extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-custom-att-script.png>)

4. Add the new custom attribute into your data feed template in:

- CATALOG
- DATA FEED MANAGER
- DATA FEED



(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-custom-att-template.png>)

An example of custom attributes:

Replace some words in the product descriptions:

```
<?
    $search=array('Nokia','BlackBerry','apple');
    $replace=array("NOKIA","BLACKBERRY","APPLE");
    return str_ireplace($search, $replace,$product->getDescription());
?>
```

Create a thumbnail of the product image:

```
<? return Mage::helper('catalog/image')->init($product, 'image')->resize(265,265); ?>
```

Filter out products that match a condition:

```
<?
    if(strlen($product->getUpc())>=12) return $product->getUpc();
    elseif ($product->getMpn()) return $product->getMpn();
    else $this->skip();
?>
```

Calculate the sum of all quantity available for a configurable product:

```
<?
    $childProducts = Mage::getModel('catalog/product_type_configurable')->getUsedProducts(null,
    $stock_count=0;
    foreach($childProducts as $child){
    $stock_count+=(int)Mage::getModel('cataloginventory/stock_item')->loadByProduct($child->get
    }
    return "<g:quantity>".$stock_count."</g:quantity> ";
?>
```

Custom options use

Data Feed Manager also allows you to create custom functions.

To create your own functions, go to:

- CATALOG
- DATA FEED MANAGER
- CUSTOM FUNCTIONS

Here is how to proceed:

1. Create a new option.



(<https://www.wyomind.com//media/extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-option-create.png>)

2. Fill in the option name with the name of your custom option.

The screenshot shows the 'New custom option' form. It has a title 'New custom option' and a sub-header 'Option configuration'. There are three main input fields: 'Option code *', 'Number of additional parameters *', and 'Custom php script *'. Each field has a small help icon to its right. The 'Option code *' field has a tooltip that says 'Use only letters (a-z), numbers (0-9) or underscore(_) in this field'. The 'Number of additional parameters *' field has a tooltip that says 'Use only numbers (0-9) in this field'. The 'Custom php script *' field has a tooltip that says 'Create your custom php script (no opening or closing tags needed)'. There are also 'Reset' and 'Save' buttons at the bottom right.

(<https://www.wyomind.com//media/extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-option-name.png>)

3. Define the number of additional parameters expected by your custom option.

New custom option

Option configuration

Option code * Use only letters (a-z), numbers (0-9) or underscore(_) in this field

Number of additional parameters * Use only numbers (0-9) in this field

Custom php script * Create your custom php script (no opening or closing tags needed)

(<https://www.wyomind.com//media/extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-option-param.png>)

4. Add your custom script - No need to add the PHP tags (<? ... ?>)

New custom option

Option configuration

Option code * Use only letters (a-z), numbers (0-9) or underscore(_) in this field

Number of additional parameters * Use only numbers (0-9) in this field

Custom php script * Create your custom php script (no opening or closing tags needed)

(<https://www.wyomind.com//media/extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento1-custom-options-newcustomoption3.png>)

5. Add the new custom option into your data feed template.



(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-option-template.png>)

An example of option functions:

Pad an attribute to a certain length with another string:

```
<?
/*
 * {{ean,[str_pad],[10],[0]}}
 * when ean is 12345, then the output is 0000012345
 */
return str_pad($self,$param[1],$param[2],STR_PAD_LEFT)
/*
 * could also be used as a inline function
 * {{ean,[str_pad($self,10,0,STR_PAD_LEFT)]}}
 */
?>
```

Format a number:

```

<?
    /*
    * {{price,[number_format],[2],[,]}}
    * when price is 9.99345, then the output is 9,99
    */
    return number_format($self,$param[1],$param[2],' ')
    /*
    * could also be used as a inline function
    * {{ean,[number_format($self,2,',',' ')]}}
    */
?>

```

Shopping engines configurations

How to configure my template with Google Shopping?

Usefull links for Google Shopping

Google Shopping allows you to create datafeed in destination of Google Shopping.

Here are some tips which will be usefull to create your datafeed.

First, we recomand you to look at these 2 links which will be helpfull:

- [Google Shopping's feed specifications](#)
- [Google Shopping's taxonomy](#)

Once your data feed has been uploaded into your Google Merchant, you can check what products have been added to the Google Shopping index :

[https://www.google.com/products?authorid=\[your Account ID\]](https://www.google.com/products?authorid=[your Account ID])

File format with Google Shopping

Google Shopping handles CSV or XML file format ([More information](#)).

The following explanations will only concern the most used method: XML files.

File names should remain consistent for all updates. Do not include the date in your file name. File name cannot contain spaces or illegal symbols.

Required attributes with Google Shopping

Required attributes with Google Shopping are depending on what kind of products you want to submit.

For complete informations about required attributes, please refer to the dedicated page on the Merchant Center .

Here are the most frequent required attributes used by Google Shopping.

(Between brackets, specific name for XML files)

	Attribute	Description
General attributes	id (<i><g:id></i>)	The unique identifier for your product (usually the SKU code)
	Title (<i><title></i>)	The name of your item (limited to 70 characters) For variant products, you are required to include the common title of the variant items.

	<p>Description (<description>)</p>	<p>The description of the item. Google Shopping recommend you submit around 500 to 1000 characters, but you can submit up to 10000 characters.</p>
	<p>Google product category (<g:google_product_category>)</p>	<p>Google's category of the item. Indicates the category of the product being submitted, according to the Google product taxonomy ↗. By example, for hard drives items: <i>Electronics &gt; Computers &gt; Computer Components &gt; Storage Devices &gt; Hard Drives</i> Please refer to our Category mapping FAQ ↗ to understand how to use it with Simple Google Shopping.</p>
	<p>Link (<link>)</p>	<p>The URL directly linking to your item's page on your website. <u>Note</u>: In addition, you may also specify an <code>adwords_redirect</code> attribute for tracking purposes.</p>
	<p>Image link (<g:image_link>)</p>	<p>The URL of the main image of the item. Google Shopping recommends images of at least 400 x 400 pixels. The accepted image formats are JPG (or JPEG), PNG, GIF, BMP and TIFF.</p>
	<p>Condition (<g:condition>)</p>	<p>The condition or state of the item. Only accepted values: <i>new, used, refurbished</i>.</p>
Prices and Availability	<p>Availability (<g:availability>)</p>	<p>The availability status of the item Only accepted values: <i>in stock, available for order, out of stock, preorder</i></p>
	<p>Price (<g:price>)</p>	<p>The price of the item. Important : - For the US, don't include tax in the price. For all other countries, value added tax (VAT) has to be included in the price. - The price must include a currency according to ISO 4217 Standard ↗. (by example : EUR for euros, USD for US Dollars, GBP for pounds). - For products on sale, you must use <code><g:sale_price></code> (see below) and use normal price in <code><g:price></code>. <u>Example</u> : <code><g:price>15.00 USD</g:price></code></p>
<p>Note about unique product identifiers : except for custom made items, you need to submit at least two of the three attributes 'brand', 'gtin' and 'mpn'.</p>		
Unique Product Identifiers	<p>Brand (<g:brand>)</p>	<p>The brand of the item. You must not provide your store name as the brand unless you manufacture the product.</p>
	<p>GTIN (<g:gtin>)</p>	<p>Global Trade Item Number (GTIN) of the item. These identifiers include UPC (in North America), EAN (in Europe), JAN (in Japan), and ISBN (for books).</p>
	<p>MPN (<g:mpn>)</p>	<p>Manufacturer Part Number (MPN) of the item. This code uniquely identifies the product to its manufacturer.</p>
Only Apparel Products	<p>Gender (<g:gender>)</p>	<p>3 accepted values: <i>male, female, unisex</i>. <u>Note</u>: Required for all apparel items in feeds that target the US, UK, DE, FR, and JP. Recommended for other countries and product categories (might become required in the future).</p>
	<p>Age group (<g:age_group>)</p>	<p>2 accepted values: <i>adult, kids</i>. <u>Note</u> : Required for all apparel items in feeds that target the US, UK, DE, FR, and JP. Recommended for other countries and product categories (might become required in the future).</p>

Shared by Apparel and Variants	color (<g:color>)	Color of the item. This defines the dominant color(s) of an item. <u>Note</u> : When a single item has multiple colors, combine them with '/' (/ with XML files) in order of prominence (dominant color first, then accent colors).
	Size (<g:size>)	Size of the item. <u>Note</u> : - required for 'clothing' and 'shoes' sub-categories within the 'apparel and accessories' category targeting the US, UK, DE, FR, and JP. - For apparel variants submitted to the US, UK, DE, FR, and JP: Each different size or size combination must be submitted as its own item, in conjunction with the 'item group id' (<g:item_group_id>).
	Item group id (<g:item_group_id>)	Shared identifier for all variants of the same product. Allows to identify all variations (size and/or color) of a same item. We recommend you to use the product'SKU parent as item group id value

Recommended attributes with Google Shopping

Here are some attributes (non-exhaustive list) recommended by Google Shopping that we think could be quite useful to add in your template:

(Between brackets, specific name for XML files)

	Attribute	Description
General attributes	Product type (<g:product_type>)	Your category of the item. This attribute also indicates the category of the product being submitted, but you can provide your own classification . You can include more than one 'product type' attribute value if products apply to more than one category.
	Additional image link (<g:additional_image_link>)	Additional URLs of images of the item. You can include up to 10 additional images per item by including the attribute multiple times.
	Sale price (<g:sale_price>)	Advertised sale price of the item. Important: - You must still include the <g:price> attribute in your feed, which must reflect the non-sale price you have previously sold the item for. - As the normal price, the sale price must include a currency according to ISO 4217 Standard ↗ . - You must not include a price value of 0.00 USD.
	Sale price effective date (<g:sale_price_effective_date>)	Used in conjunction with sale price, this attribute indicates the date range during which the sale price applies. <u>Information about time format:</u> Start and end dates separated by a forward slash (/). The start is specified by the format (YYYY-MM-DD), followed by the letter T, the time of the day when the sale starts, followed by an expression of the timezone for the sale. The end date should be in the same format. <u>Example:</u> 2011-03-01T13:00-0800/2011-03-11T15:30-0800
Tax and Shipping	Tax (<g:tax>)	Value added tax (VAT). Providing this attribute for an item overrides the default tax costs specified in your Google Merchant Center settings. <u>Note:</u> This attribute is only accepted in the US, if your feed targets a country outside of the US, please do not use this attribute.

	<p>Shipping (<g:shipping>)</p>	<p>Shipping cost. Providing this attribute for an item overrides the global shipping settings you defined in your Google Merchant Center settings. More informations about shipping and tax ↗</p>
	<p>Shipping weight (<g:shipping_weight>)</p>	<p>The weight of the product used to calculate the shipping cost of the item. If you have specified a global shipping rule that is dependent on shipping weight, this attribute will be used to calculate the shipping cost of the item automatically. Accepted weight units: <i>lb, oz, g, kg</i>.</p>
AdWords attributes	<p>Note : These attributes help you to organize your AdWords campaigns and track the traffic to your website.</p>	
	<p>Adwords grouping (<g:adwords_grouping>)</p>	<p>Used to group products in an arbitrary way. It can be used for Product Filters to limit a campaign to a group of products, or Product Targets to bid differently for a group of products. Required if you want to bid differently to different subsets of products in the CPC or CPA % version. It can only hold one value. <u>Example:</u> <code><g:adwords_grouping>comforters</g:adwords_grouping></code></p>
	<p>Adwords labels (<g:adwords_labels>)</p>	<p>Very similar to <code><g:adwords_grouping></code>, but it will only work on CPC. It can hold multiple values, allowing a product to be tagged with multiple labels. <u>Example:</u> <code><g:adwords_labels>clothing</g:adwords_labels></code> <code><g:adwords_labels>shoes</g:adwords_labels></code></p>
	<p>Adwords redirect (<g:adwords_redirect>)</p>	<p>Allows advertisers to specify a separate URL that can be used to track traffic coming from Google Shopping. <u>Example:</u> <code><g:adwords_redirect>https://www.merchant.com/product.html</g:adwords_redirect></code></p>

Example of template with XML format with Google Shopping

Here is an example of template in XML format:

```

<g:id>{sku}</g:id>
  <title>{name,[substr],[70],[...]}</title>
  <link>{url}</link>
  <!-- you must change the currency unit or convert the price to match with your magento store and y
  <g:price>{normal_price,[USD]} USD</g:price>
  {G:SALE_PRICE,[USD]}
  <g:online_only>y</g:online_only>
  <description>{description,[html_entity_decode],[strip_tags]}</description>
  <g:condition>new</g:condition>
  {G:PRODUCT_TYPE,[10]}
  {G:GOOGLE_PRODUCT_CATEGORY}
  {G:IMAGE_LINK}
  <g:availability>{is_in_stock?[in stock]:[out of stock]}</g:availability>
  <g:quantity>{qty}</g:quantity>
  <g:featured_product>{is_special_price?[yes]:[no]} </g:featured_product>
  <g:color>{color,[implode],[,]}</g:color>
  <g:shipping_weight>{weight,[float],[2]} kg</g:shipping_weight>
  {G:PRODUCT_REVIEW}
  <g:manufacturer>{manufacturer}</g:manufacturer>
  <!-- In most of cases brand + mpn are sufficient, eg. :-->
  <g:brand>{manufacturer}</g:brand>
  <g:mpn>{sku}</g:mpn>
  <!-- But it is better to use one of these identifiers if available : EAN, ISBN or UPC, eg : -->
  <g:gtin>{upc}</g:gtin>

```

How to create a data feed in Google Merchant Center?

In order to submit your products to Google Shopping, you'll have to create a data feed in your Google Merchant Center [↗](#). Once you're login into Google Merchant Center, go in the **Data feeds** section.



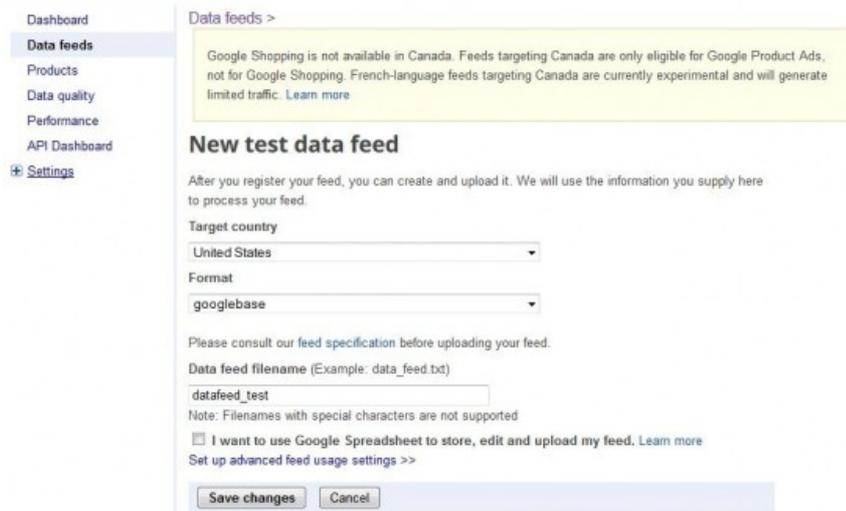
(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-data-feeds.jpg>)

You can choose between creating a *New Data Feed* and a *New Test Data Feed*.

We recommend that you start by creating a *New Test Data Feed*.

After clicking on **New Test Data Feed**, you have to choose the target country and a name for your data feed ("datafeed_test" by example).

You can confirm by clicking on **Save changes**.



(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-new-data-feed.jpg>)

Now you can see your data feed appears in the data feeds list.



(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-datafeed-list.jpg>)

Next step will be to setup this data feed so it will import the data feed you created with the extension Simple Google Shopping in your Magento store.

Click on the link **Create schedule**. You are now accessing the schedule upload options.

You can manage the Google Merchant's importation frequency. You can choose to import you data feed monthly, weekly or daily at a specific time.

Paste your data feed's URL in the field **Feed URL**.

(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-planning.jpg>)

Click on **Schedule and fetch now** to start the importation and the process of your data feed. Your data feeds' status shows **Processing...**

File	Upload	Last upload date	Status
datafeed_test Settings	Daily upload Manual upload	Mar 25, 2013 1:43 pm CET	Processing... Test Feed [?]

(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-processing.jpg>)

After a few minutes, Google Merchant Center shows you the number of inserted items and offers you to display errors in your data feed.

File	Upload	Last upload date	Status
datafeed_test Settings	Daily upload Manual upload	Mar 25, 2013 1:43 pm CET	33 of 122 items inserted (View errors) Expires Apr 24, 2013 Test Feed [?] Load into debugger [?]

(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-processing-finish.jpg>)

Click on **View errors** will show you errors in your data feed.

Google Merchant Center discerns 2 kinds of anomaly: errors and warnings.

As long it will remain errors in your data feed, Google Merchant will not display your items in Google Shopping.

It's very important to read errors' descriptions. It will help you to understand and fix quickly your data feed's issues.

Once the errors fixed, you'll be able to proceed to the creation of a new data feed (not a test one this time) which will perform the processing and referencing of your items by Google Shopping.

Warning: the validation of the first submission of your data feed to Google can take some days because it's performing manually by Google's teams. The next submissions will be validated almost instantly.

How to configure my template with Shopping.com?

Data Feed Manager allows you to create datafeed in destination of Shopping.com.

Here are some tips which will be useful to create your datafeed.

First, we recommend you to look at the following link which will be helpful:

- [Shopping.com's feed specifications](#) 

File format for Shopping.com

Shopping.com handle CSV or XML file format.

For CSV files, data should be enclosed within double quotation marks ("), and tab as field separator.

The following characters are forbidden within any cell:

- Smart Quotes Character ("?)
- Carriage Returns Character
- New Line Characters
- Any character whose ascii value is less than 32, e.g. Carriage Returns (ascii 10), Newline (ascii 13)
- HTML, CSS, or any other formatting languages, e.g. "\n\r", <p>,
.

File name should remain consistent for all updates. (e.g. MerchantnameSDC.csv). Do not include the date in your file name.

File name cannot contain spaces or illegal symbols.

Required attributes for Shopping.com

These 9 attributes are required. Products without these values will not display on Shopping.com :

(Between brackets, specific name for XML files)

Attribute	Description
Unique Merchant SKU (<i>Merchant_SKU</i>)	The unique SKU number you use internally to identify this item
Product Name (<i>Product_Name</i>)	Product Names must be clear and concise. They should also be descriptive of the product being sold.(limited to 90 characters)
Product URL (<i>Product_URL</i>)	URL on your website containing the product detail and buy button for the applicable product
Image URL (<i>Image_URL</i>)	Provide the URL to the item's largest primary image. Minimum size accepted: 150 pixels on at least one side of the image. Any modern format is acceptable, i.e. JPEG, PNG, GIF, TIFF
Current Price (<i>Current_Price</i>)	Provide your current selling price for the item before tax or shipping
Stock Availability (<i>Stock_Availability</i>)	Denote Y or YES to show you have the product in stock and N, NO or NONE when the product is out of stock. Out-of-stock products will not be shown on Shopping.com. (Allowed values : <i>In Stock, Yes, Y, 1, Limited availability, Out of Stock, No, N, None, Pre-Order, Back-Order</i>)
Condition (<i>Condition</i>)	Please use this field to indicate if the item is "New", "Used" or "Refurbished". IMPORTANT: Shopping.com currently displays "New" products only. "Used" and "Refurbished" products are rejected automatically
These 2 last attributes are only required for media	
MPN/ISBN (<i>MPN ou ISBN</i>)	The ISBN is a unique machine-readable identification number. Shopping.com supports both the 10 and 13 digit versions of ISBN

**UPC ou EAN
(UPC ou EAN)**

Shopping.com supports both the 12-digit Universal Product Codes (UPC) and the 13-digit EAN codes. UPC or EAN are strongly recommended in all categories

Recommended attributes for Shopping.com

Here are some attributes (non-exhaustive list) recommended by Shopping.com we think are quite useful to add in your template :
(Between brackets, specific name for XML files)

Attribute	Description
Shipping Rate (Shipping_Rate)	Shipping cost for this item. Use 0 to denote free shipping. Refer to the Merchant Account Center at PRODUCTS > EDIT-SHIPPING-INFORMATION page if you want to define Shipping Rates by Price ranges
Original Price (Original_Price)	If your website displays a price drop and/or percent savings for this item, provide here the item's original price
Brand / Manufacturer (Brand ou Manufacturer)	Brand, manufacturer, or publisher of the product
Product Description (Product_Description)	Product Descriptions should elaborate on Product Names and provide all relevant product details (limited to 4000 characters)
Product Type (Product_Type)	Provide an ordered list of nouns describing the type of product. Use a comma to separate them. Please refer to Shopping.com' specifications to find a list of relevant categories
Category (Category)	Please provide a breadcrumb describing the classification of the respective product. (ie: " ELECTRONICS > DIGITAL CAMERAS "). <i>Remember to use category mapping in Data Feed Manager to fill-in this attribute correctly</i>
Category ID (Category_ID)	Use the Shopping.com's taxonomy to provide the right Shopping.com's Category ID for this item
Parent SKU (Parent_SKU)	If the product is one variant in a family of similar products, provide here a SKU number that uniquely identifies the family
Parent Name (Parent_Name)	The name of the parent product

Example of template for Shopping.com with XML format

Here is an example of template for a website selling t-shirts:

Header:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
  <Products>
```

Pattern:

```
<Product>
  <Merchant_SKU>{sku}</Merchant_SKU>
  <Product_Name>{name,[substr],[70],[...]}</Product_Name>
  <Product_URL>{url parent}</Product_URL>
  <Image_URL>{image parent}</Image_URL>
  <Current_Price>${price,[USD],[0]}</Current_Price>
  <Shipping_Rate>$.50</Shipping_Rate>
  <Stock_Availability>{is_in_stock parent?in stock:[out of stock]}</Stock_Availability>
  <Condition>New</Condition>
  <Original_Price>${normal_price,[USD],[0]}</Original_Price>
  <Manufacturer>{brand}</Manufacturer>
  <Product_Description>{short_description parent,[html_entity_decode],[strip_tags]}</Product_Descrip
  <Product_Type>Clothing</Product_Type>
  <Category>{category_mapping,[0]}</Category>
  <Category_ID>31515</Category_ID>
  <Parent_SKU>{sku parent}</Parent_SKU>
  <Parent_Name>{name parent,[substr],[70],[...]}</Parent_Name>
  <Estimated_Ship_Date>Allow 10 days for Delivery</Estimated_Ship_Date>
  <Gender>{gender}</Gender>
  <Color>{color,[implode],[,]}</Color>
  <Size>{size,[implode],[,]}</Size>
  <Alternative_Image_URL_1>{image,[1]}</Alternative_Image_URL_1>
  <Alternative_Image_URL_2>{image,[2]}</Alternative_Image_URL_2>
  <Alternative_Image_URL_3>{image,[3]}</Alternative_Image_URL_3>
  <Alternative_Image_URL_4>{image,[4]}</Alternative_Image_URL_4>
  <Alternative_Image_URL_5>{image,[5]}</Alternative_Image_URL_5>
  <Merchandising_Type>New</Merchandising_Type>
  <Bundle>No</Bundle>
</Product>
```

Footer:

```
</Products>
```

How to configure my template with Bing Shopping?

Data Feed Manager allows you to create datafeed in destination of Bing Shopping.

Here are some tips which will be useful to create your datafeed.

First, we recommend you to look at these 2 links which will be helpful:

- [Getting started with Bing Shopping](#)
- [Bing Merchant Center](#)

File format for Bing Shopping

Bing Shopping doesn't support XML file format. You should create TXT file format.

HEADER (the line with attribute's names) is required by Bing Shopping.

Data should be enclosed without any characters, and you should use tab as field separator.

The following characters are forbidden within any cell:

- Smart Quotes Character ("")
- Carriage Returns Character
- New Line Characters
- Any character whose ascii value is less than 32, e.g. Carriage Returns (ascii 10), Newline (ascii 13)
- HTML, CSS, or any other formatting languages, e.g. "\n\r", <p>,

- No \$ sign in the price values and no lbs in shipping weights (only numbers, limited to 2 decimal places)

File name should remain consistent for all updates. Do not include the date in your file name.

File name cannot contain spaces or illegal symbols.

Bing Shopping recommends to use **bingshopping.txt** as a file name.

Required attributes for Bing Shopping

Bing Shopping requires these 6 attributes:

Attribute	Description
MerchantProductID	You can either use internal Magento ID, either product'SKU as a unique identifier
Description	Detailed description of your offer (limited to 5000 characters). No HTML coding. Ensure no descriptions are wrapped in quotes. Please do not include promotional text.
ProductURL	URL of the product. Link to your website where a potential buyer can complete the purchase of your product.
ImageURL	Provide the URL to the item's largest primary image. Acceptable formats: BMP,JPG, PNG, GIF, TIFF and EXIF
Price	Provide your current selling price for the item before tax or shipping
Title	Product Names must be clear and concise (limited to 255 characters)

Recommended attributes for Bing Shopping

Bing Shopping highly recommends to use these attributes in your template :

Attribute	Description
Brand	The product manufacturer's name, brand name or publisher's name
MPN	The ISBN is a unique machine-readable identification number. Bing Shopping supports both the 10 and 13 digit versions of ISBN
UPC	Universal Product Code. Only 8 and 12 digit codes are supported
ISBN	International Standard Book Number
SKU	The SKU unique number of the product. Used to refer to different versions of the same product
Availability	The current availability for the offer. Choose only one of following values: In Stock; Out of Stock; Pre-Order; Back-Order
Shipping	This attribute will override any info set in the Shipping tab of your Bing Shopping account. Each shipping attribute group must be separated with a comma and the four sub-attributes by colons such as: Country:Region:Service:Price . Only Price value is required. Three colons are required even for blank values. <u>Examples:</u> No shipping world wide = :::0 No shipping US wide = US:::0 Multiple variables are allowed and are split with a comma = US:024*:Ground:7.95,US:MA:Ground:5.95
MerchantCategory	Your internal category hierarchy for this product
ShippingWeight	The weight of the product in pounds. Don't use lbs sign and avoid the use of leading zeros

Condition	<p>Acceptable condition values are: New; Used; Collectable; Open Box; Refurbished; Remanufactured</p> <p>WARNING: Currently Bing Shopping shows only new products. If not specified, Bing Shopping assumes the product is new</p>
Tax	<p>This attribute will override any info set in the Tax tab of your Bing Shopping account. Each tax attribute group must be separated with a comma and the four sub-attributes by colons such as: Country:Region:Rate:Tax_Ship.</p> <p>Only Rate value is required.</p> <p>Three colons are required even for blank values.</p> <p><u>Examples:</u></p> <p>No tax world wide = ::0:</p> <p>No tax US wide = US::0:</p> <p>Multiple variables are allowed = US:CA:8.25:y,US:926: *:8.75:y</p>
B_Category	<p>Bing Shopping Category.</p> <p>Bing's list of categories to match your product to Bing's standardized structure.</p> <p><i>Remember to use category mapping in Data Feed Manager to fill-in this attribute correctly</i></p>

Example of template for Bing Shopping with TXT format

Here is an example of template:

Edit data feed 'BingShopping' [Back] [Reset] [Delete] [Save & Generate] [Save & Continue] [Copy] [Save]

Configuration

Feed name * BingShopping

File type * txt

Path * /feeds/

File status * enabled

Store View * English

Extra header
(for Amazon Data Feeds)

Include header * yes

Columns name	Pattern
MerchantProductID	{id}
Title	{name}
Brand	{manufacturer}
MPN	{sku}
UPC	{upc}
ISBN	{isbn}
SKU	{sku}
ProductURL	{url parent[{subst}],2000}
Price	{price}
Availability	{is_in_stock parent?}{in Stock}
Description	{description parent}{html_ent
ImageURL	{image}
Shipping	US:\$ 99
MerchantCategory	{categories .1}
B_Category	{category_mapping}
ShippingWeight	{weight}{float}{2}
Condition	new
Tax	US:0:

Add field

Fields delimiter * \tab

Fields enclosure none

(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-bingshopping.png>)

How to configure my template with Shopzilla?

File format for Shopzilla

Shopzilla doesn't support XML file format. You should create TXT file format.

HEADER (the line with attribute's names) is required by Shopzilla.

Data should be enclosed without any characters, and you should use tab as field separator.

The following characters are forbidden within any cell:

- Smart Quotes Character ("?)
- Carriage Returns Character
- New Line Characters
- Any character whose ascii value is less than 32, e.g. Carriage Returns (ascii 10), Newline (ascii 13)
- HTML, CSS, or any other formatting languages, e.g. "\n\r", <p>,

File name should remain consistent for all updates. Do not include the date in your file name.

File name cannot contain spaces or illegal symbols.

Required attributes for Shopzilla

Shopzilla requires 15 attributes in the datafeed, but only 5 must contain data. Others are highly recommended by Shopzilla.

WARNING: these 15 attributes must respect this order:

1. **Category ID**
2. Manufacturer
3. **Title**
4. Description
5. **Product URL**
6. Image URL
7. **SKU**
8. Availability
9. Condition
10. Ship Weight
11. Ship Cost
12. Bid
13. Promotional Code
14. UPC
15. **Price**

Bold attributes above must contain data.

Here are the 5 required attributes :

Attribute	Description
Category ID	Shopzilla Category. You have to map your own categories by using Shopzilla's taxonomy ↗ . <i>Remember to use category mapping in Data Feed Manager to fill-in this attribute correctly</i>
Title	Product Names must be clear and concise (limited to 100 characters)
Product URL	URL of the product. Link to your website where a potential buyer can complete the purchase of your product.
SKU	The SKU unique number of the product. Used to refer to different versions of the same product
Price	Provide your current selling price for the item before tax or shipping (don't use \$ sign)

Recommended attributes for Shopzilla

Shopzilla highly recommends to use these attributes in your template :

Attribute	Description
Manufacturer	The product manufacturer's name, brand name or publisher's name. Leave empty if you don't know the manufacture's name.
Description	Detailed description of your offer (limited to 1000 characters). No HTML coding. Ensure no descriptions are wrapped in quotes. Please do not include promotional text.
Image URL	Provide the URL to the item's largest primary image. Acceptable formats: JPG or GIF. Image size should be at least 200x200 pixels and no greater than 1000x1000. Image URLs must begin with https:// (Shopzilla doesn't support secure locations (https://))
Availability	Lets your consumers know if the product is currently available on your website. Choose only one of following values: In Stock, Back-Order, Pre-Order, See Site, Out of Stock, Limited Qty, Special Order. If empty, Shopzilla will default to "See Site for Availability".
Condition	Designates the state of the product. Acceptable condition values are: New, Refurbished, Used, Open Box, OEM. If empty, Shopzilla will default to "New".
Ship Weight	Designates the weight of your product (in pounds).
Ship Cost	Designates a flat shipping cost for the product, if desired. This field is not necessary if you plan to use the Shopzilla Shipping Tool to specify ship costs. Note: To designate free shipping, please use 0.00
Bid	This is an optional field and recommended only for those merchants programmatically setting their product bids. Please refer to Feed Specifications for further information about this attribute.
Promotional Code	This field is used to add promotional text next to your product. Please refer to Feed Specifications for further information about this attribute.
UPC	Universal Product Code. Only 8 and 12 digit codes are supported

Example of template for Shopzilla with TXT format

Here is an example of template:

Configuration

Feed name *

File type *

Path *

File status *

Store View *

Extra header
(for Amazon Data Feeds)

Include header *

Columns name	Pattern	
<input type="text" value="Category ID"/>	<input type="text" value="{category_mapping.[0]}"/>	<input type="button" value="X"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
<input type="text" value="Manufacturer"/>	<input type="text" value="{manufacturer}"/>	<input type="button" value="X"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
<input type="text" value="Title"/>	<input type="text" value="{name}"/>	<input type="button" value="X"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
<input type="text" value="Description"/>	<input type="text" value="{description.[html_entity_dec...}"/>	<input type="button" value="X"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
<input type="text" value="Product URL"/>	<input type="text" value="{url}"/>	<input type="button" value="X"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
<input type="text" value="Image URL"/>	<input type="text" value="{image.[0]}"/>	<input type="button" value="X"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
<input type="text" value="SKU"/>	<input type="text" value="{sku}"/>	<input type="button" value="X"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
<input type="text" value="Availability"/>	<input type="text" value="{is_in_stock?[in stock]}[out of"/>	<input type="button" value="X"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
<input type="text" value="Condition"/>	<input type="text" value="new"/>	<input type="button" value="X"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
<input type="text" value="Ship Weight"/>	<input type="text" value="{weight}"/>	<input type="button" value="X"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
<input type="text" value="Ship Cost"/>	<input type="text" value="0,00"/>	<input type="button" value="X"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
<input type="text" value="Bid"/>	<input type="text"/>	<input type="button" value="X"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
<input type="text" value="Promotional Code"/>	<input type="text"/>	<input type="button" value="X"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
<input type="text" value="UPC"/>	<input type="text" value="{ean}"/>	<input type="button" value="X"/> <input type="button" value="↑"/> <input type="button" value="↓"/>
<input type="text" value="Price"/>	<input type="text" value="{price}"/>	<input type="button" value="X"/> <input type="button" value="↑"/> <input type="button" value="↓"/>

Fields delimiter *

Fields enclosure

(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-shopzilla.png>)

How to configure my template with Amazon?

Data Feed Manager allows you to create datafeed in destination of Amazon.

Here are some tips which will be usefull to create your datafeed.

First, we recommend you to look at this link which will be helpfull:

Amazon Browse Tree Guide (BTG) (Amazon's specifications) [↗](#)

For Amazon Product Ads, we recommend the following documentation :

<https://sellercentral.amazon.com/gp/help/> [↗](#)

File format with Amazon

Amazon doesn't support XML file format. You should create TXT file format.

HEADER (the line with attribute's names) is required by Amazon.

ATTENTION : Amazon requires an **extra-header** , which allows Amazon to identify which kind of products is in you data feed, and the actual used version. Here is an example concerning Home category products:

```
TemplateType=Home Version=2012.0228 This row for Amazon.com use only. Do not modify or delete.
```

In order to use these specific extra-header, Data Feed Manager has a specific field named **Extra header** :



(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-amazon-extraheader.png>)

Data should be enclosed without any characters, and you should use tab as field separator.

The following characters are forbidden within any cell:

- Smart Quotes Character ("?")
- Carriage Returns Character
- New Line Characters
- Any character whose ascii value is less than 32, e.g. Carriage Returns (ascii 10), Newline (ascii 13)
- HTML, CSS, or any other formatting languages, e.g. "\n\r", <p>,

File name should remain consistent for all updates. Do not include the date in your file name.

Informations groups with Amazon

Amazon distincts different informations groups to describe the products you'll send them in your data feed. Some groups are generic and suitable for all types of products:

- Basic Product information
- Offer Information
- Sales Price information
- Item discovery information
- Image Information
- FBA (Fulfillment By Amazon)
- Variation information (if your Product is available in different variations)
- Product Dimensions
- Infrequently used attributes

Beside these generic groups, there are specific groups depending of your products' categories. Here are specific groups from **Home** category:

- Product specific attributes included in the "ArtAndCraft"
- Product specific attributes included in the "Kitchen"
- Product specific attributes included in the "Indoor Home"
- Product specific attributes included in the "Outdoor Home and Garden"
- Battery Information

The very first thing to do before creating your Amazon data feed is to identify which kind of products your Magento catalog contains. Once this is done, you'll have to refer to the Amazon Browse Tree Guide (Amazon BTG) matching this product category to create your data feed in the best way.

By example, if you're selling furnitures, you'll have to refer to **Home** Amazon BTG, which matches with following themes:

- Art and Craft
- Kitchen
- Indoor Home
- Outdoor Home and Garden

We'll use this particular example for following explanations.

Required attributes with Amazon

Usually, Amazon requires these 10 attributes in the datafeed.

WARNING: These 10 attributes have to be in the data feed unless Amazon will reject it.

Attribute	Description
SKU	The SKU unique number of the product. Used to refer to different versions of the same product
StandardProductID	A standard, alphanumeric string that uniquely identifies the product. This could be a Global Catalog ID (GCID), UPC, EAN or GTIN.
StandardIDType	The type of standard, unique identifier entered in the ProductID field. Select one of the following options: Global Catalog ID (GCID), UPC, EAN or GTIN.
ProductName	Product Names must be clear and concise (limited to 500 characters). Don't use special characters.
ProductType	Identify the product types. With our example : For all Indoor products, choose from: Kitchen, BedAndBath, ArtAndCraft, Furniture or the general category Product type of Home. Select one of the following options: Art, BedAndBath, FurnitureAndDecor, Home, Kitchen, OutdoorLiving, SeedsAndPlants.
Brand	The brand or manufacturer of the product. Populate this field so your brand name is displayed on the Amazon.com site. (Brand or Manufacturer is required, both are desired)
Manufacturer	Specify the manufacturer for your product. (Brand or Manufacturer is required, both are desired)
ItemType	Use this to specify what your item is. This field is primarily used for browse classification. Refer to the Browse Tree Guide (BTG) ☞ for allowed values. <i>Remember to use category mapping in Data Feed Manager to fill-in this attribute correctly</i>
ItemPrice	The price at which the merchant offers the product for sale, expressed in U.S. dollars. Don't use currency unit.
Link	The URL of the product. Must start with https://

Recommended attributes with Amazon

Amazon highly recommends to use these attributes in your template :

Detailed description of your offer (limited to 2000 characters). Don't use special characters.

Attribute	Description
Currency	The currency code indicating what currency the pricing data is in. (USD for US Dollars, EUR for Euro, GBP for Pounds).
ShippingWeight	The weight of the product when packaged to ship. A number with up to 10 digits to the left of the decimal point and 2 digits to the right of the decimal point.
ShippingWeightUnitOfMeasure	The unit of measure used to describe the weight of the product: OZ, LB, GR or KG.
MainImageURL	Provide the URL to the item's largest primary image. Preferred format: JPG. Image size must be at least 500 pixels in either height or width and should be 1000 pixels or larger. Image URLs must begin with https:// .

OtherImageURL1- OtherImageURL8	Additional images showing different sides of a product, the product in use, or details that aren't visible in the MAIN image.
Quantity	The quantity of the item you are making available for sale. Quantity is required for all "child" or stand-alone items. If left blank when first submitting the product information, the item will be displayed as "Out of stock".

How to configure my template with Price Grabber?

Data Feed Manager allows you to create datafeed in destination of Price Grabber.

Here are some tips which will be useful to create your datafeed.

First, we recommend you to look at these 2 links which will be helpful:

[Price Grabber Integration Guide](#)

[Price Grabber's taxonomy](#)

File format with Price Grabber

Price Grabber doesn't support XML file format. You should create TXT file format.

HEADER (the line with attribute's names) is required by Price Grabber.

Data should be enclosed without any characters, and you should use tab as field separator.

The following characters are forbidden within any cell:

- Smart Quotes Character ("?")
- Carriage Returns Character
- New Line Characters
- Any character whose ascii value is less than 32, e.g. Carriage Returns (ascii 10), Newline (ascii 13)
- HTML, CSS, or any other formatting languages, e.g. "\n\r", <p>,

File name should remain consistent for all updates. Do not include the date in your file name.

File name cannot contain spaces or illegal symbols.

Required attributes with Price Grabber

Price Grabber requires some attributes regarding to the kind of product you're selling in your store.

By example, Home & Personal Products requires 9 attributes, where Tech Products & Software only requires 7 attributes (not the same as well).

Please have a look at the [Price Grabber Integration Guide](#) to know exactly which attributes are required for your products.

However, here are the most current required attributes:

Attribute	Description
Retsku	The SKU unique number of the product. Used to refer to different versions of the same product
Product Title	Product Names must be clear and concise (limited to 100 characters)
Detailed Description	Detailed description of your offer (limited to 1500 characters). No HTML coding. Ensure no descriptions are wrapped in quotes. Please do not include promotional text.
Categorization	Price Grabber Category. You have to map your own categories by using Price Grabber's taxonomy . By example : laptops products needs to have: <ul style="list-style-type: none"> ● COMPUTERS > LAPTOPS & ACCESSORIES > LAPTOPS as matching category. <i>Remember to use category mapping in Data Feed Manager to fill-in this attribute correctly</i>
Product URL	URL of the product. Link to your website where a potential buyer can complete the purchase of your product.

Primary Image URL	Provide the URL to the item's largest primary image. Image URLs must begin with https:// (Price Grabber doesn't support secure locations (https://))
Selling Price	Current selling price of the item. Please omit currency symbols. Prices should be provided in the appropriate currency (US Dollars for PriceGrabber.com, Pound Sterling for PriceGrabber.co.uk, Canadian Dollars for PriceGrabber.ca).
Manufacturer Name	The product manufacturer's name, brand name or publisher's name. Leave empty if you don't know the manufacturer's name.

Recommended attributes with Price Grabber

Price Grabber highly recommends to use these attributes in your template :

Attribute	Description
Manufacturer Part Number	Manufacturer-issued part number (MPN) for the product.
UPC / EAN	UPC (US, Canada): Unique code of 12 numbers. EAN (Europe): Unique code of 13 numbers.
Availability	Lets your consumers know if the product is currently available on your website. Choose only one of following values: Yes, No ou Preorder
Weight	The product's shipping weight. Required if you select a shipping cost formula which depends on weight. The unit for weight is pounds in the US, and kilograms for UK.
Shipping Cost	Required if you select "Use the shipping costs provided in my product data feed" in the Shipping section of Price Grabber Login.

How to configure my template with Nextag?

File format with Nextag

Nextag handles TXT file format.

HEADER (the line with attribute's names) is required by Nextag.

Data should be enclosed without any characters, and you should use pipe (|) as field separator.

The following characters are forbidden within any cell:

- Smart Quotes Character ("")
- Carriage Returns Character
- New Line Characters
- Any character whose ascii value is less than 32, e.g. Carriage Returns (ascii 10), Newline (ascii 13)
- HTML, CSS, or any other formatting languages, e.g. "\n\r", <p>,

File name should remain consistent for all updates. Do not include the date in your file name.

File name cannot contain spaces or illegal symbols.

Required attributes with Nextag

Here are required attributes with Nextag:

Attribute	Description
-----------	-------------

Category: Nextag Numeric ID	The Nextag category of the products you are listing. You have to map your own categories by using Nextag's taxonomy ↗ . By example : laptops products needs to have: 300359 as matching category. <i>Remember to use category mapping in Data Feed Manager to fill-in this attribute correctly</i> Note : You must match one of the "Category" columns to a column in your feed.
Category: Other Format	The category of the products you are listing. These should be the categories you use on your site. <u>Example</u> : COMPUTERS > LAPTOPS. Note : You must match one of the "Category" columns to a column in your feed.
Click-Out URL	URL of the product.
Condition	The condition of the products you are selling. Choose only one of following values: New, Open Box, Refurbished, Pre-Owned, Like New, Good, Very Good, Acceptable
Description	Detailed description of your offer. No HTML coding. Ensure no descriptions are wrapped in quotes. Please do not include promotional text.
Ground Shipping	The amount you charge for ground shipping. In dollars with or without the \$ sign. If you use a carrier such as FedEx or UPS, you can define this on your Seller Dashboard to ensure you display accurate shipping.
Image URL	The URL or address of the images of your products. URL starting with https://, https:// or just www and ending with .jpg or.gif
ISBN	The ISBN of any book products you are selling. This field is only required if you are selling books.
MUZE ID	The MUZE IDs of any music, video or video game products you are selling. This field is only required if you are selling music, video or video game products.
Price	The price of the items you are selling. In dollars with or without the \$ sign.
Product Name	The name of the products you are selling.
Seller Part #	The SKU unique number of the product.
Stock Status	The availability or stock status of your products. Choose only one of following values: In Stock, Yes, le nombre en stock, Out of Stock, <blank>. If you leave this blank, Nextag will assume the product is in stock.
Weight	The weight of your products. Number in pounds (lbs).

Recommended attributes with Nextag

Nextag highly recommends to use these attributes in your template :

Attribute	Description
Cost-per-Click	The Cost-per-Click or CPC that you want to pay Nextag when a buyer clicks from our site to yours. In dollars with or without the \$ sign. This should only be used if you want to bid on individual products using your Product File.

Distributor ID	The distributor ID of your products. One of the following initials: Ingram (I, IM, IP, IA), TechData (T), eNighted (E), ISBN, UPC, LCC plus the number ID.
Ingram Part #	The Ingram ID of your products.
Manufacturer	The name of the manufacturer of your products.
Manufacturer Part #	The manufacturer's part number for the product. It is strongly recommended that you provide this information to improve the quality of your listings on Nextag.
Marketing Message	The text message you want to display on our site next to your display name. Example : Free Tax and Shipping
UPC	The Universal Product Code (UPC) of the products you are selling.
ListPrice	The item's price when it is not on sale. In dollars with or without the \$ sign

How to configure my template with Twenga?

Data Feed Manager allows you to create datafeed in destination of Twenga.

Here are some tips which will be useful to create your datafeed.

First, we recommend you to look at this link which will be helpful:

Twenga's feed specifications [↗](#)

File format with Twenga

Twenga handles CSV/TXT or XML file format.

About CSV/TXT files :

- data shouldn't be enclosed
- you can use either tab, semicolon, comma or pipe as field separator.

About XML files :

- the datafeed must be UTF-8 encoded (<?xml version="1.0" encoding="UTF-8" ?>)
- **catalog** tags (<catalog>...</catalog>) must enclose the content.
- each product must be in **product** tags (<product>...</product>)

The following characters are forbidden within any cell:

- Smart Quotes Character ("?")
- Carriage Returns Character
- New Line Characters
- Any character whose ascii value is less than 32, e.g. Carriage Returns (ascii 10), Newline (ascii 13)
- HTML, CSS, or any other formatting languages, e.g. "\n\r", <p>,
.

File name should remain consistent for all updates. (e.g. MerchantnameSDC.csv). Do not include the date in your file name.

File name cannot contain spaces or illegal symbols.

Required attributes with Twenga

These 7 attributes are required. Products without these values will not display on Twenga :

Attribute	Description
product_url	URL on your website containing the product
designation	Product Names must be clear and concise. (limited to 255 characters)

price	Final price of the product, with or without VAT, depending on local regulations.
category	Navigation path of the product (Breadcumb). Default separator : > (greater). The category mapping of the product (by example : " <ul style="list-style-type: none"> ● HOME>TELEVISIONS>LCD TVS>THOMSON"). <i>Remember to use category mapping in Data Feed Manager to fill-in this attribute correctly</i>
image_url	Main picture of the product. Minimum size : 500x500 pixels. Maximum size : 2000x2000 pixels.
description	Product's description. Only the first 180 characters will be display in Twenga
regular_price	Product's sale price, before any sale. Leave empty if the product doesn't have any discount.

Recommended attributes with Twenga

Here are recommended attributes by Twenga :

Attribute	Description
merchant_id	Product's ID. The internal product ID used by the merchant.
manufacturer_id	The manufacturer ID. You may use SKU for this attribute.
shipping_cost	Shipping cost. If not specified, leave empty or specify "NC" For free shipping, specify "0".
in_stock	Denote Y to show you have the product in stock and N when the product is out of stock.
availability	Number of products available
condition	For new products, use "0". For used or refurbished products, use "1".
upc_ean	UPC or EAN of the product.
brand	Brand or manufacturer of the product.

Example of template with XML, CSV and TXT format with Twenga

Here is an example of template with XML format:

```

<?xml version="1.0" encoding="utf-8"?>
<catalog>
  <product>
    <product_url>http://www.mywebsite.com/products/TV-LCD-THOMSON-32HE8234B.html</product_url>
    <designation>TV LCD THOMSON 32HE8234B</designation>
    <price>369.00</price>
    <category>Home/Television/LCD TVs/Thomson</category>
    <image_url>http://www.mywebsite.com/products/1/32/324805268_F.jpg</image_url>
    <description>Format 16/9 screen size 32" [...] 2 bass reflex speakers</description>
    <brand>Thomson</brand>
    <merchant_id>2148</merchant_id>
    <manufacturer_id>THL324805268</manufacturer_id>
    <shipping_cost>0</shipping_cost>
    <in_stock>Y</in_stock>
    <stock_detail>49</stock_detail>
    <condition>0</condition>
    <isbn></isbn>
    <upc_ean>324805268</upc_ean>
    <product_type>1</product_type>
  </product>
  <product>
    <product_url>http://www.mywebsite.com/books/The-Third-Policeman-9781564782144.html</product_url>
    <designation>The Third Policeman</designation>
    <price>10.04</price>
    <category>Home/Books/Fiction Books/Fiction/Literature</category>
    <image_url>http://www.mywebsite.com/books/4/12/9781564782144.jpg</image_url>
    <description>Originally submitted and rejected [...] after the author's death.</description>
    <brand>Naxos</brand>
    <merchant_id>BK9781564782144</merchant_id>
    <manufacturer_id>9781564782144</manufacturer_id>
    <shipping_cost>3.99</shipping_cost>
    <in_stock>Y</in_stock>
    <stock_detail>3</stock_detail>
    <condition>1</condition>
    <isbn>9781564782144</isbn>
    <upc_ean></upc_ean>
    <product_type>4</product_type>
  </product>
</catalog>

```

(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-twenga-xml.png>)

Here is an example of template with CSV format:

product_url	designation	price	category	image_url	description	brand	merchant_id	manufacturer_id	shipping_cost	in_stock	stock_detail	condition	isbn	upc_ean	product_type
http://www.TV LCD THOMSON 369.00	Home/Telev	http://www.Format 16/9	Thomson	2148	THL324805268	0	Y	49	0				324805268	1	
http://www.The Third Po 10.04	Home/Book	http://www.Originally su	Naxos	BK978156478	9781560000000	3.99	Y	3	1	9781560000000				4	

(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-twenga-csv.png>)

Here is an example of template with TXT format:

```

product_url; designation; price; category; image_url; description; brand; merchant_id; manufacturer_id; shipping_cost; in_stock; stock_d
http://www.mywebsite.com/products/TV-LCD-THOMSON-32HE8234B.html; TV LCD THOMSON 32HE8234B; 369.00; Home/Televiions/LCD TVs/Thom
http://www.mywebsite.com/books/The-Third-Policeman-9781564782144.html; The Third Policeman; 10.04; Home/Books/Fiction Books/Fict

```

(<https://www.wyomind.com//media//extensions/data-feed-manager/magento-1/userguide/data-feed-manager-magento-twenga-txt.png>)

How to configure my template with Idealo?

Data Feed Manager allows you to create datafeed in destination of Idealo.

Here are some tips which will be usefull to create your datafeed.

First, we recommand you to look at these 2 links which will be helpfull:

Idealo's technical informations [↗](#)

Idealo's sample CSV file [↗](#)

Here are the admission requirements for Idealo:

- Your store contains at least 20 products already existing in Idealo's catalog.
- All prices are displayed in euro (including VAT)
- Your legal infos are correct
- Your policies are accessible and understandable to any customer
- Customers can choose between at least 2 payment methods.
- Shipping won't exceed 15.00 for small appliances
- Shipping won't exceed 50.00 for big appliances

File format with Idealo

Idealo is used to work with CSV files.

Data shouldn't be enclosed

You can use either tab, semicolon or pipe as field separator.

The CSV file must contain a header line describing each column's names.

Do not include the date in your file name.

File name cannot contain spaces or illegal symbols.

Required attributes with Idealo

These attributes are required by Idealo.

Products without these values will not display on Idealo:

Attribute	Description
Catégorie	Internal category path. (By example: digital photography - digital camera - compact digital camera - Canon - Canon Digital Ixus 70)
Marque / Fabricant	The brand or manufacturer of the product
Produit	The name of the product
Référence fabricant (HAN)	The manufacturer reference
Prix en euro (TVA incluse)	Price in euro (including VAT)
Frais de port	Shipping cost. If free shipping, fill in with "0".
URL produit	URL on your website containing the product
Délai de Livraison	Delivery time / availability
EAN (European article number)	EAN code of the product
Code CIP	Only for pharmaceuticals
ISBN	Only for books
Ecotaxe	Eco-tax amount, only if not already included in the price

Recommended attributes with Idealo

Here are recommended attributes by Idealo:

Attribute	Description
Numéro d'article (interne)	Internal reference of the product (usually SKU)
information sur le produit	The description of the product (short version if available)
URL photo	The URL of the main picture of the product.
Prix Promo	Sale price of the product

Example of template with CSV format with Idealo

Here is an example of template with CSV format:

Numéro d'article (interne);EAN (European article number);Référence fabricant (HAN);Marque / Fabricant;Produit;
 40101966;3165140580113;GSB 14.4-2-LI;FBC BOSCH;PERCEUSE VISSEUSE 14,4V 1,3AH LI-ION;Outils élé
 40101915;0000401019151;BHP453RHEX;MAKITA;PACK PERCEUSE A PERCUSSION 18V 1,5AH LXT;Perceuses, visse
 43302205;3148518627948;CPE.PB;FACOM;LOT DE 2 PINCES "PACK PREMIER";Pincés standards et coupantes;
 46600252;0000466002525;1-94-747;STANLEY TOOLS;SERVANTE DE CHANTIER LOCK & STOCK FATMAX;Boite à out
 45900043;3253561956294;1-95-629;STANLEY TOOLS;GILET PORTE-OUTILS FATMAX XTREM;Saccosches et porte-c
 44800848;9999999999999;0-10-825;STANLEY TOOLS;COUTEAU CUTTER PLIABLE FATMAX;Cutteurs et ciseaux;
 46600415;3253561974755;1-97-475;STANLEY TOOLS;JEU 2 TRETEAUX PLIABLES COMPACTS;Etats et tréteaux de
 60800432;3253561951480;1-95-151;STANLEY TOOLS;LAMPE TORCHE LED PETIT MODÈLE;Torches, ampoules, et
 42606588;3253560975456;0-97-500;STANLEY TOOLS;CLE A MOLETTE A CLIQUET FATMAX;Clés de serrage à ouv

How to configure my template with eBay?

File format with eBay

eBay handles CSV/TXT or XML file format.

About CSV/TXT files :

- data must be enclosed with double quotation marks ("")
- If you intend to use the double quote character (") itself and not as a delimiter, it should be repeated. For example, '15" TV' should be entered as ""15"" TV".
- you must use comma as separator in CSV files, and either tab, semicolon, comma or pipe in TXT files.

About XML files :

- the datafeed must be UTF-8 encoded (<?xml version="1.0" encoding="UTF-8" ?>)
- each product must be in **product** tags (<product>...</product>)

The following characters are forbidden within any cell:

- Smart Quotes Character ("?)
- Carriage Returns Character
- New Line Characters
- Any character whose ascii value is less than 32, e.g. Carriage Returns (ascii 10), Newline (ascii 13)
- HTML, CSS, or any other formatting languages, e.g. "\n\r", <p>,
.

File name should remain consistent for all updates. (e.g. ebay.csv). Do not include the date in your file name.

File name cannot contain spaces or illegal symbols.

Required attributes with eBay

These 9 attributes are required by eBay. Products without these values will not display on eBay.
 (between brackets, the name to use with XML files)

Attribute (Merchant_SKU)	Description (Merchant_SKU)
Unique Merchant SKU (Merchant_SKU)	The unique SKU number you use internally to identify this item.
Product Name (Product_Name)	Product Names must be clear and concise. They should also be descriptive of the product being sold. Use the following template to build your product names: [MAIN BRAND] [PRODUCT NAME] [GENERIC PRODUCT TYPE] Example : <i>Leica M7 Rangefinder 35mm Camera</i>
Product URL (Product_URL)	URL on your website containing the product detail and buy button for the applicable product.

Image URL <i>(Image_URL)</i>	URL to the item's largest primary image. Minimum size accepted: 150x150 pixels.
Current Price <i>(Current_Price)</i>	The current selling price for the item before tax or shipping.
Stock Availability <i>(Stock_Availability)</i>	Denote In Stock to show you have the product in stock and Out of Stock when the product is out of stock.
Condition <i>(Condition)</i>	Please use this field to indicate if the item is "New", "Used" or "Refurbished". IMPORTANT: eBay Commerce Network currently displays "New" products only. "Used" and "Refurbished" products are rejected automatically.
MPN/ISBN (media only) <i>(MPN ou ISBN)</i>	The ISBN is a unique machine-readable identification number. eBay Commerce Network supports both the 10 and 13 digit versions of ISBN.
UPC/EAN (media only) <i>(UPC ou EAN)</i>	eBay Commerce Network supports both the 12-digit Universal Product Codes (UPC) and the 13-digit EAN codes. UPC or EAN are strongly recommended in all categories. Items with UPC or EAN will be better classified in eBay Commerce Network.

Recommended attributes with eBay

Here are recommended attributes by eBay:
(between brackets, the name to use with XML files)

Attribute <i>(Merchant_SKU)</i>	Description <i>(Merchant_SKU)</i>
Shipping Rate <i>(Shipping_Rate)</i>	Shipping cost for this item. Use 0 to denote free shipping.
Original Price <i>(Original_Price)</i>	The item's original price.
Brand / Manufacturer <i>(Manufacturer)</i>	Brand, manufacturer, or publisher of the product.
Product Description <i>(Product_Description)</i>	Product Descriptions should elaborate on Product Names and provide all relevant product details. Limited to 4000 characters.
Product Type <i>(Product_Type)</i>	Provide an ordered list of nouns describing the type of product. Example: "Sneaker, Cross Trainer".
Category <i>(Category)</i>	Please provide a breadcrumb describing your classification of the respective product (use > to delimiter sub-categories)
Category ID <i>(Category_ID)</i>	Please use the updated Category worksheet to provide the right eBay Commerce Network's Category ID for this item. (cf. "Categories" tab in eBay's specifications). Remember to use category mapping in Data Feed Manager to fill in this attribute properly.
Parent SKU <i>(Parent_SKU)</i>	If the product is one variant in a family of similar products, provide here the SKU of the parent product.
Parent Name <i>(Parent_Name)</i>	If the product is one variant in a family of similar products, provide here the name of the parent product.
Estimated Ship Date <i>(Estimated_Ship_Date)</i>	Provide your estimated time frame for the shipping of the product. Example: <i>Allow 2 - 3 weeks for delivery.</i>
Gender <i>(Gender)</i>	Indicate the gender(s) for whom the product is intended. Use a comma to separate them. Admitted values: <i>Men, Women, Boys, Girls, Infants and Toddlers, Unisex</i>
Color <i>(Color)</i>	The primary colors for the product. In the event that the product is multi-colored, please separate the colors with a slash (/). Please provide an individual record for each color.

Material (Material)	Indicate the materials from which the product is intended. Values should be comma separated.
Size (Size)	The size of the product.
Age Range (Age_Range)	eBay strongly recommends that age ranges are included for data feeds that include toys, apparel, jewelry and sporting goods. Examples: <i>3 - 6 months, 5 years and up, Adult,...</i>

Example of template with XML format with eBay

Here is an example of template with XML format:

```
<Products>
  <Product>
    <Merchant_SKU>93345</Merchant_SKU>
    <Product_Name>J. Crew White Floral Summer Sundress</Product_Name>
    <Product_URL>https://www.abcxyz123.com/store/product.html</Product_URL>
    <Image_URL>https://www.abcxyz123.com/images/6857623495-1000.jpg</Image_URL>
    <Current_Price>$99.99</Current_Price>
    <Shipping_Rate>5.95</Shipping_Rate>
    <Stock_Availability>Pre-Order</Stock_Availability>
    <Condition>New</Condition>
    <UPC>053759061260</UPC>
    <EAN></EAN>
    <MPN></MPN>
    <ISBN></ISBN>
    <Original_Price>$199.99</Original_Price>
    <Manufacturer>J. Crew</Manufacturer>
    <Product_Description>Medium weight floral printed sundress. Perfect for Sunday brunch.</Product_De
    <Product_Type>Dress > Sun Dress</Product_Type>
    <Category>Clothing & Accessories > Women clothing > Dress > Sundress</Category>
    <Category_ID>31515</Category_ID>
    <Parent_SKU>93344</Parent_SKU>
    <Parent_Name>J. Crew Floral Summer Sundress</Parent_Name>
    <Estimated_Ship_Date>Allow 3-4 Weeks for Delivery</Estimated_Ship_Date>
    <Gender>Women, Girls</Gender>
    <Color>White</Color>
    <Material>Cotton, Spandex</Material>
    <Size>10</Size>
    <Age_Range></Age_Range>
  </Product>
</Products>
```