



MASS PRODUCT IMPORT & UPDATE

User Guide



Mass Product Import & Update allows you to update or import massively products in Magento in a record time thanks to CSV and XML files available from your website's server, from remote servers through FTP or HTTP.

Mass Product Import & Update allows you to create and/or update the below product attributes for all store views or specific store views:

- All system attributes (SKU, type, attribute set, visibility, status)
- All product attributes created within Magento (input, dropdown, yes/no, multi select, text...)
- All product image attributes (image gallery, base image, thumbnail, small images...)
- All stock attributes (qty, availability, backorder, manage stock, low stock notification...)
- Categories associated with products
- Prices, special prices, special price dates, tax class...

How to configure Mass Product Import & Update

Configure Mass Product Import & Update in a general way

General Settings

To configure *Mass Product Import & Update* in a general way, you have to go in:

[STORES](#) [SETTINGS](#) [CONFIGURATION](#) [WYOMIND](#) [MASS PRODUCT IMPORT & UPDATE](#)

Log file

Enable log

Allows you to activate the log file available under the var/log directory. The file is named *Wyomind_MassProductImportAndUpdate.log*.

Enable log [global]

Note the log file will be written only if the primary log file is active in:

[STORES](#) [SETTINGS](#) [CONFIGURATION](#) [ADVANCED](#) [DEVELOPER](#) [LOGS SETTINGS](#)

Mapping preview

Number of rows in mapping preview

Allows you to define the number of rows returned in the browser data of the preview when editing a profile.

Number of rows in mapping preview [global]

Cron job reporting

With *Mass Product Import & Update*, you have the possibility to enable cron job reporting.

■ Enable cron job reporting

Allows you to define whether you want to enable the reporting or not in order to get notified each time a profile runs and if an error happens. The more important is the number of rows, the more the load time increases. A good compromise could be 1000 rows.

■ Sender Email

Type the email of the sender.

■ Sender Name

Define the name of the sender.

■ Send the reports to

Specify the email addresses to which the report must be sent. Emails must be separated with a comma (,).

■ Report title

Define your own title for the report.

Enable cron job reporting [global] Yes

Sender Email [store view] test@email.com

Sender Name [store view] John Do

Send the reports to [global] test1@email.com,test2@email.com,test3@email.com
Emails must be separated with a comma (,).

Report title [global] Mass Product Import & Update

Cron interval

By default, cron tasks can be scheduled each period of 30 minutes.

If you need a more precise hour, you can change that interval in the **Cron interval** field.

Cron Interval [global]

Create an import profile with Mass Product Import & Update

With *Mass Product Import & Update*, you can update Magento products or create new ones.

For this, edit one of the sample profile patterns or create a new profile by clicking from:

- [🔍 SYSTEM](#)
- [🔍 DATA TRANSFER](#)
- [🔍 MASS PRODUCT IMPORT & UPDATE](#)

Mass Product Import & Update > Profiles 🔍 🔔 👤 wyomind

[Import a profile](#) [Create a new profile](#)

Search [Reset Filter](#) 4 records found 20 per page < 1 of 1 >

ID	Profile name	File location	File type	Method	Status	Last execution	Action
1	xml_product_update	Url	xml	Update products only	PENDING	Nov 8, 2018, 12:45:23 AM Show report	Edit Delete Run profile Show report
2	Configurable Products On The Fly	Url	csv	Update products and import new products	PENDING	Nov 23, 2018, 3:09:14 AM Show report	
3	Simple + Configurable Products	Url	csv	Update products and import new products	PENDING	Nov 23, 2018, 3:19:16 AM Show report	
4	xml_inventory_update	Url	xml	Update products only	PENDING	Nov 8, 2018, 12:45:23 AM Show report	

Black Box

The black-box is a tool that will help you configure your mapping seamlessly. It is composed of 3 major features.

Input Data

To get a preview of the input data, click on .

xml_product_update - Source Preview

identifier	name	description	price
<code>\$cell[0] \$cell['identifier']</code>	<code>\$cell[1] \$cell['name']</code>	<code>\$cell[2] \$cell['description']</code>	<code>\$cell[3] \$cell['price']</code>
24-MB01	Joust Duffle Bag	The sporty Joust Duffle Bag can't be beat - not in the gym, not on the luggage carousel, not anywhere. Big enough to haul a basketball or soccer ball and some sneakers with plenty of room to spare, it's ideal for athletes with places to go. Dual top handles. Adjustable shoulder strap. Full-length zipper. L 29" x W 13" x H 11".	34.00
		Convenience is next to nothing when your day is crammed with action. So whether you're heading to class, gym, or the unbeaten path, make sure you're not	

Input data Library Output data

Library

The library allows you to check what are the values you can use for each attribute. You will also find some examples to help you.

xml_product_update - Library

Identifier	Values	
Attribute	Type	Example
Required attributes Sku	Unique Identifier	(empty)
Required attributes Attribute set	Attribute set id or Attribute set name (case insensitive)	-
Required attributes Type	Simple, Configurable, Downloadable, Virtual, Bundle, Grouped, ...	-
Required attributes Website	Website id or Website name (case insensitive)	-
Required attributes Tax Class	Taxclass id or Taxclass name (case insensitive)	-
Required attributes Visibility	Product visibility id or product visibility name (case insensitive)	-
Required attributes Status	Product Status (enabled/disabled)	true, yes, in stock, enable, enabled or false, no, out of stock, disable, disabled

Input data Library Output data

Output Data

Clicking on will give you a preview of the data as they'll be updated.

xml_product_update - Output Preview

Identifier	Product Name	Description	Price
24-MB01	Joust Duffle Bag	The sporty Joust Duffle Bag can't be beat - not in the gym, not on the luggage carousel, not anywhere. Big enough to haul a basketball or soccer ball and some sneakers with plenty of room to spare, it's ideal for athletes with places to go. Dual top handles. Adjustable shoulder strap. Full-length zipper. L 29" x W 13" x H 11".	34.00
24-MB04	Strive Shoulder Pack	Convenience is next to nothing when your day is crammed with action. So whether you're heading to class, gym, or the unbeaten path, make sure you've got your Strive Shoulder Pack stuffed with all your essentials, and extras as well. Zippered main compartment. Front zippered pocket. Side mesh pocket. Cell phone pocket on strap. Adjustable shoulder strap and top carry handle.	32.00
		The Crown Summit Backpack is equally at home in a gym locker, study cube or a pup tent, so be sure yours is packed with books, a bag lunch, water bottles, yoga block, laptop, or	

➔ Input data Library Output data ➔

Settings

In this section, you will be able to define the source of the data.

Profile settings

Define the settings for your profile.

Profile Name

Only alphabetic and number are allowed + hyphens (-) and underscores (_).
Profile's name must be unique.

SQL mode

When SQL mode is enabled, no data are updated or imported. Running the profile will only produce a SQL file. This file could be executed directly in your database manager.

If you activate SQL mode, you'll need to fill in several fields:

SQL file name

Name of the SQL file to generate. For example:

`XML_sample.sql`

SQL file path

Is the path where the SQL file will be generated (relative to Magento root folder). For example:

`var/sample`

Profile method

You can choose the action that the profile will trigger:

Update products only

Import new products only

Update products and import new products

Filter lines

Leave empty to export all lines:

e.g:

means all lines will be imported

Type the numbers of the lines you want to import, making sure to separate each line or range with a comma (,):

e.g: 2,4,6,8,10

means lines 2,4,6,8,10 will be imported

Use a dash (-) to denote a range of lines:

e.g: 8-10

means lines 8,9,10 will be imported

Use a plus (+) to import all lines from a line number:

e.g: 4+

means all lines from line 4 will be imported

Use regular expressions surrounded by # to indicate a particular group of identifiers to import:

e.g: #ABC-[0-9]+#

means all lines with an identifier matching the regular expression will be imported

Note you can combine all those options. For example **2,6-10,15+** means line **2,6,7,8,9,10,15,16,17,...** will be imported.

Import Profile Settings

Profile name *

SQL mode * When SQL mode is enabled, no stocks are updated. Running the profile will only produce a SQL file. This file could be executed directly in your database manager

SQL file name * Name of the SQL file to generate.

SQL file path * Path where the SQL file will be generated (relative to Magento root folder).

Profile method

Filter lines

When the **Profile method** option is on **Update products and import new products**, in the **Mapping & Rules** tab, you can define for each Target attribute dropdown if the value applies to the new products, or/and to the existing products:

- New products only
- Existing products only
- Both, new and existing products

Description

- New products only
- Existing products only
- Both, new and existing products

You can then click Input data from the blackbox to see your source file

File Location

This section allows you to specify where the source data file is stored. You have several options:

File Location

Allows you to define where is stored the file that must be used for the profile:

Magento File System: The file is stored in the Magento root directory or a subdirectory of the Magento root directory.

FTP server: The file is stored on a remote server available from FTP or SFTP. Do not prefix the domain name or the IP with FTP://. You'll be able to check if the FTP connexion is valid.

Url: The file is stored on a remote server available from HTTP or HTTPS.

Webservice: The file is available on a remote server accessible as web service. The extension uses the CURL PHP library to connect the web service. You can provide the web service a login/password couple and any parameters formatted as url parameters, as follows:

parameter1=value1¶meter2=value2¶meter3=value3

Dropbox: You can generate your token from your Dropbox account

Path to File

The path to the file including the file name and the extension as it appears in the file system of the server.

Magento file system: File path relative to Magento root folder.

Ftp server: File path relative to ftp user root folder.

Url: Url of the file.

Web service: Url of the web service.

Dropbox: Url of the dropbox service

File Location

File location *

Drag files from your desktop
txt, csv or xml files only
Maximum size 512Mb

Path to file *

- **Magento file system** : File path relative to Magento root folder
- **Ftp server** : File path relative to ftp user root folder
- **Url** : Url of the file
- **Web service** : Url of the web service
- **Dropbox** : Url of the dropbox service

Note you can drag and drop files from your desktop. Only TXT, CSV or XML files are allowed.

File Type

Allows you to specify the file type used for the update, XML or CSV.

■ File type

You have the choice between CSV and XML files.

For CSV files, define:

■ Column separator

Define the column separator.

■ Text delimiter

Select the text delimiter.

■ The first line is a header

Specify when the first line is a header (CSV files only).

■ Magento export file

Magento default export files are made of empty rows with values related to main rows, activating this option will merge the values into the main rows.

File Type

File type *

Column separator

Text delimiter

The first line is a header

Magento export file *

For XML files, define:

■ Xpath to products

In case of an update from an XML file, fill in the XPath that targets the products. The XPath is case sensitive.

■ XML structure

You have two options; the **Automatic detection** or the **Predefined structure**. The automatic detection fits for simple files made of only one nesting level such as the above file sample.

■ Predefined structure

If you've selected the **Predefined structure**, you have the possibility to define your own file structure. When the structure of your file is quite complex, the info may be dispatched in deep nodes and sometimes in tag attributes such as:

```
url="something"
```

Below is an example of complex file where you need a predefined structure:

```
<offers xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1">
  <offer price="165.07" msrp="189.00" availability="3">
    <name><![CDATA[Amazing product]]></name>
    <images>
      <img url="https://www.example.com/path/to/product/image/1.jpg" />
      <img url="https://www.example.com/path/to/product/image/2.jpg" />
    </images>
    <videos />
    <categories>
      <category><![CDATA[Category A]]></category>
      <category><![CDATA[Category B]]></category>
      <category><![CDATA[Category C]]></category>
    </categories>
    <description><![CDATA[Amazing product description]]></description>
    <attributes>
      <attribute name="sku"><![CDATA[XYZ]]></attribute>
      <attribute name="ean"><![CDATA[123456789]]></attribute>
      <attribute name="tax"><![CDATA[VAT 20%]]></attribute>
    </attributes>
  </offer>
</offers>
```

The XPath that targets the products in the above file is `/offers/offer`

The JSON syntax you need to use is:

```
{
  "sku": "attributes/attribute[@name='sku']",
  "ean": "attributes/attribute[@name='ean']",
  "price": "@price",
  "msrp": "@msrp",
  "tax": "attributes/attribute[@name='tax']",
  "name": "name",
  "description": "description",
  "category_1": "categories/category[1]",
  "category_2": "categories/category[2]",
  "category_3": "categories/category[3]",
  "image_1": "images/img[1]/@url",
  "image_2": "images/img[2]/@url"
}
```

File Type

File type *

Xpath to products *
xPath where the product data is stored in the XML file, e.g./catalog/products/product

XML structure *
The automatic detection of the XML structure fits for simple files made of only one nesting level.

Predefined XML structure

```
{
  "sku": "attributes/attribute[@name='sku']",
  "ean": "attributes/attribute[@name='ean']",
  "price": "@price",
  "msrp": "@msrp",
  "tax": "attributes/attribute[@name='tax']",
  "name": "name",
  "description": "description",
  "category_1": "categories/category[1]",
  "category_2": "categories/category[2]",
  "category_3": "categories/category[3]",
  "image_1": "images/img[1]/@url",
  "image_2": "images/img[2]/@url"
}
```

The predefined XML structure must be a valid json string made of a key/value list that define the column names and the XPath associated to the column

DIRECTIONS FOR USE IN THE PREDEFINED XML STRUCTURE FIELD

- Use a valid **Json string** made of a key/value list that define the column names and the Xpath associated to the columns.
- Use **@** to access the attribute values. For example:
`@price`
- Use **[@attribute=value]** to access a node from which the attribute is value. For example:
`[@name='ean']`
- Use **[number]** to access another node that has the same name. For example:
`category`
- Use **Xpath axis** to represent a relationship to the current node, and locate nodes. For example:
`child::item`

Post Process Action

You have the possibility to define an action after the file is processed.

■ **Action**

Do Nothing
Delete the import file
Move import file

■ **Move to folder**

Define the file path where you want to move the import file, relative to your Magento root folder.

Note that this option is available only for update files locally hosted in your Magento File System.

■ **Run indexers**

You can choose to run the indexers after the import or not:

No: no indexer will run and you will have to reindex manually from the CLI

Only the required indexers: automatically decides which indexers must run

Only the selected indexers: let you decide which indexers to run

If the profile takes too much time to load, you can disable the indexers.

■ **Indexer to run**

Choose the indexers to run:

catalogrule_rule
catalogrule_product
catalog_product_price
cataloginventory_stock
inventory
catalog_product_attribute
catalog_url
catalogsearch_fulltext
catalog_category_product
catalog_product_category

Catalog_product_flat and catalog_category_flat are managed if the option is enabled in:

⊙ STORES ⊙ SETTINGS ⊙ CONFIGURATION ⊙ CATALOG ⊙ CATALOG ⊙ STOREFRONT

Post Process Action

Action on import file *

Move to folder *
File path relative to Magento root folder

Run indexers *
The indexes may need to be updated after importing data
- **No**: no indexer will run and will have to reindex manually from the CLI
- **Only the required indexers**: automatically decides which indexers must run
- **Only the selected indexers**: let you decide which indexers to run

Indexer to run *

- catalogrule_rule
- catalogrule_product
- catalog_product_price
- cataloginventory_stock
- catalog_product_attribute
- catalog_url
- catalogsearch_fulltext
- catalog_product_category

Advanced Settings

System Settings

■ Automatic action for missing products

Define what to do for each product that is missing from the data file. You have several options:

Do Nothing

Disable the product

Delete permanently the product

Mark the product as out of stock

■ Target

When a product is imported using *Mass Product Import & Update*, it is associated with the profile. If you've selected an action for missing products, you'll need to choose a target:

Only products related to current profile: products that are imported/updated with the same profile.

Only products not related to current profile: all the products that are not created/updated by the profile.

All products

System Settings

Automatic action for missing products
Action for each product that is missing from the data file

Target
Target of products impacted by the action

Stock Settings

Automatic stock status update

You can choose to automatically update the stock status (in stock / out of stock).

Stock Settings

Automatic stock status update

Stock status will be automatically updated (in stock / out of stock)

Image Settings

Indicates the path to the directory in which the images to import are stored.

■ Images location

- Http server (url)
- Magento File System
- Ftp server

Depending on what you've selected as the images location, you'll need to fill in several fields.

For the FTP server:

- Use SFTP
- Use active mode
- Host
- Port
- Login
- Password

For the FTP Server and the Magento file System:

- Path to images directory

Image Settings

Images location *
Use SFTP *

Use active mode *

Host

Port

Login

Password

[Test Connection](#)

Path to images directory

Category Settings

Mass Product Import & Update allows you to create categories on the fly.

■ **Create categories on the fly**

Yes or No.

■ **New categories active by default**

Yes or No.

■ **New categories included in menu by default**

Yes or No.

■ **Category tree auto-detection**

To automatically detect the category tree.

You need to use the complete path to the category. The category levels must be separated with slashes: A/B/C.

For example, instead of "Shirts", you should write:

Default Category/Men/Shirts

When the **Category tree auto-detection** is disabled, the category is added to the root category defined in **New categories are children of**.

■ **New categories are children of**

Choose the parent category.

Category Settings

Create categories on the fly * ▼

New categories active by default * ▼

New categories included in menu by default * ▼

Category tree auto-detection * ▼
Category levels must be separated by slashes (/)

New categories are children of * ▼

When the categories associated with the products are updated, then all previous products and categories associations are deleted.

Configurable Product Settings

Create parent of configurable products on the fly

You can also create/update a configurable product from the simple products available in the source file by enabling this option.

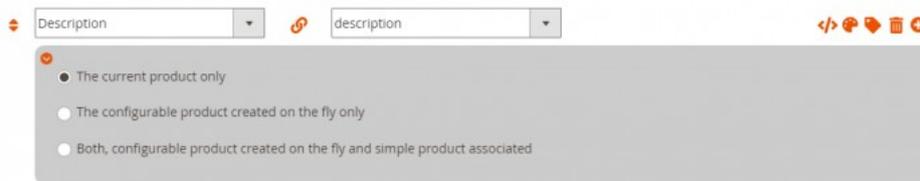
ConfigurableProduct Settings

Create parent of configurable products on the fly * ▼

Enable that option only if the configurable products don't exist in the source file and if you want to create/update these products from the simple products information.

When that option is enabled, in the **Mapping & Rules** tab, you can find for each Target attribute dropdown an option for the configurable products to create on the fly. You have the choice to apply the settings on:

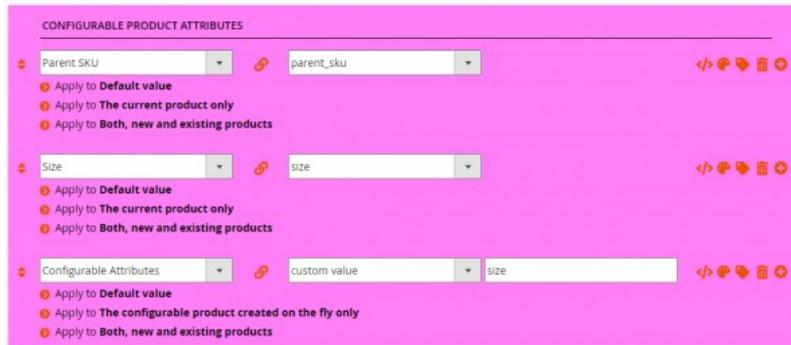
- The current product only
- The configurable product created on the fly only
- Both, configurable product created on the fly and simple product associated



System attributes of the configurable products created on the fly must receive a value either by mapping a field from the source file **for the simple product and the configurable product** or **for the configurable product only**.

So that the configurable product and the simples variants are linked, the following attributes must be mapped:

- the SKU of the parent product or the SKU of the children product
- the configurable attributes



Note that if you want to update or import a product whether it is a configurable or not, you need to use standard attributes.

Mapping & Rules

In this section you can map each column of the file with one of the stock attributes.

Column Mapping

Each column can be mapped with one of the stock attributes. Note that at any time, you can preview the input and output data in the black box directly.

■ Target Attribute

Select the target attribute from the ones available in your system (the first line is where you define the identifier of your products). You can find all the attributes grouped by type in our documentation.

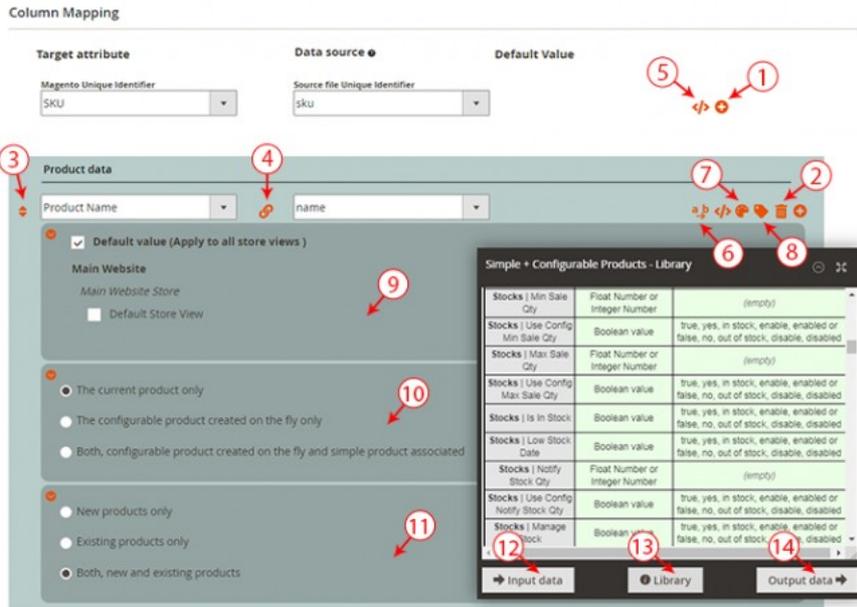
■ Data source

In the Data source, you can define the attribute source that matches with the target attribute from your source file.

■ Default value

Sometimes, some columns can be missing. For example, when updating all stocks, the file does not necessarily include all stock attributes (backorders, minimum quantity in the cart...). In that case, you can choose custom value in the Data source and apply a default value.

QUICK OVERVIEW OF THE INTERFACE



1. Add as many columns as you want by clicking on . In many cases, the update file may contain more columns than necessary; you don't need to add these columns and match them.
2. a row easily.
3. Change the order of each row using the drag and drop.
4. Deactivate a line while conserving the settings by clicking on the symbol.
5. Add your own custom script . These scripts can be previewed in the output. Note that when there is a script filled in, is underlined.
6. Add your own replacement rule clicking on the icon. These rules can be previewed in the output. Note that when there is a rule selected, is underlined
7. Apply the color of your choice to the row.
8. Apply a label to the row.
9. Define to which storeview(s) you want to apply these settings by clicking .
10. Define to which product(s) you want to apply these settings by clicking .
11. Define to which product(s) you want to apply these settings by clicking .
12. Preview your file by clicking .
13. Find all the attributes grouped by type in the .
14. Check the data as they'll be updated by clicking .

Scheduled tasks

You can schedule the profile to be executed for all days of the week and with a default interval of 30 minutes between each execution.

Settings

Days of the week	Hours of the day			
	00:00 AM	00:30 AM	00:00 PM	00:30 PM
Monday	01:00 AM	01:30 AM	01:00 PM	01:30 PM
Tuesday	02:00 AM	02:30 AM	02:00 PM	02:30 PM
Wednesday	03:00 AM	03:30 AM	03:00 PM	03:30 PM
Thursday	04:00 AM	04:30 AM	04:00 PM	04:30 PM
Friday	05:00 AM	05:30 AM	05:00 PM	05:30 PM
Saturday	06:00 AM	06:30 AM	06:00 PM	06:30 PM
Sunday	07:00 AM	07:30 AM	07:00 PM	07:30 PM
	08:00 AM	08:30 AM	08:00 PM	08:30 PM
	09:00 AM	09:30 AM	09:00 PM	09:30 PM
	10:00 AM	10:30 AM	10:00 PM	10:30 PM
	11:00 AM	11:30 AM	11:00 PM	11:30 PM

You have the possibility to change the cron task interval.

Export and import your profiles pattern

Mass Product Import & Update also allows you to import and export patterns into and from your Magento 2 admin.

Export

To export a profile, edit an existing feed from:

[SYSTEM](#) [DATA TRANSFER](#) [MASS PRODUCT IMPORT & UPDATE](#)

Edit a profile and click on .

You should find the profile pattern in your Downloads folder.

Import

Before anything, to import a profile pattern, it must be a *.conf* file, which is specific to the *Mass Product Import & Update* extension.

Click on from:

[SYSTEM](#) [DATA TRANSFER](#) [MASS PRODUCT IMPORT & UPDATE](#)

Browse the *.conf* file and click on .

The profile pattern should have been added to the list.

Note that exporting a profile pattern is different than running a profile. When you export a profile, only the template is saved, while when you run a profile, the products data are involved.

Run your Mass Product Import & Update profiles

From the profiles grid

To run the profile from the grid, you just need to click on from the action menu on the right column. Then the status will be updated with a progress bar.

ID	Profile name	File location	File type	Method	Status	Last execution	Action
1	xml_sample	Magento file system	xml	Update products and import new products	PENDING	Sep 25, 2017, 12:03:41 AM Show report	Edit Delete Run profile Show report

From the profile page

Edit the profile and then click on .



Once the import/update process has ended, a message appears with details about how many products have been updated/imported and how many products have been ignored.

With the cron tasks

When the scheduled tasks are configured for the profile, then the profile will be automatically run at the day and the time selected.

So that the scheduled tasks work well in *Mass Product Import & Update*, the main cron task in Magento must be configured properly.

You will receive an email each time the cron task is run if you have enabled the reporting in:

[STORES](#) [SETTINGS](#) [ONFIGURATION](#) [WYOMIND](#) [MASS PRODUCT IMPORT & UPDATE](#)

With the Command Lines Interface

Mass Product Import & Update includes command lines available from any shell console.

To run the profile:

```
bin/magento wyomind:massproductimport:run [-p|--profile_ids["..."]]
```

For example, for profile 2, run:

```
bin/magento wyomind:massproductimport:run -p 2
```

To execute the SQL requests that generate when the profile is set to use the SQL mode:

```
bin/magento wyomind:massproductimport:sql [-p|--profile_ids["..."]]
```

For example,, for profile 4, run:

```
bin/magento wyomind:massproductimport:sql --profile_ids=4
```

Use SQL option

Updating or importing products massively may keep your server busy some times, especially when the update or the import includes numerous products and/or many attributes.

For better performances *Mass Product Import & Update* splits the process into two distinct steps:

1. **Source file analysis:** the data are collected and the SQL queries for the update/import are built.
2. **The SQL file execution:** the data base is updated with the new data.
This step may be executed asynchronously when SQL option is enabled.

Generate the SQL file

If you prefer to generate the SQL file without executing it, you must enable the SQL mode in your profile configuration.

Then run the profile by clicking on .

Execute the SQL file

Once the file is generated in the working directory, you only need to update the database with the generated file from the command line as follows:

```
mysql -h host -p DbPassword -u DbLogin DatabaseName < path/to/mysql/file.sql
```

Advanced use

With *Mass Product Import & Update*, several events are triggered:

- **massupdateimport_start**
Triggered when the update/import process starts.
- **massupdateimport_success**
Triggered when the profile is run and the data successfully updated/imported.

- **massupdateimport_failure**

Triggered when the profile is not run and the data are not updated/imported.

The events can be used in your custom observers.

The argument passed is an instance of *Wyomind/MassProductImport/Model/Profile*.

Below are some examples of use in an observer:

- From *etc/events.xml*:

```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="../../../../../../../1
  <!-- Standard checkout -->
  <event name="massupdateimport_success">
    <observer name="editor_module_massupdateimport_success" instance="EditorModuleObserverLassUpdateImport
  </event>
</config>
```

- From *app/code/Editor/Module/Observer/MassUpdateImportSuccess.php*:

```
<?php
namespace EditorModuleObserver;
class LassUpdateImportSuccess implements MagentoFrameworkEventObserverInterface
{
    public function execute(MagentoFrameworkEventObserver $observer){
        $profile= $observer->getEvent()->getProfile();
        // TODO: Implement your own logic.
    }
}
```

Attribute specifications to import or to update

System attributes

The below attributes are required when importing new products.

If these attributes are not available in the mapping, then the imported products will be visible neither in the front-end or the back-end.

The system attributes have a global scope and cannot be updated/imported specifically for each store views.

SKU

The **unique identifier** of the product to update or to import.

For example:

sku1

Attribute set

The attribute set to apply to the product.

You can either use **the attribute set name** (case insensitive) or **the attribute set ID**.

For example:

bag

15

To know the attribute sets available in your system, go to:

[STORES](#) [ATTRIBUTES](#) [ATTRIBUTE SETS](#)

Type

The product type.

You must use **the product type name** (case insensitive).

It can be:

- simple
- configurable
- grouped
- bundle
- virtual
- downloadable
- custom product type

For example:

simple

Website

The website(s) to which belongs the product.

You can either use **the website name** (case insensitive) or **the website ID**.

For example:

Main Website Store

1

To know the names and the ids of the websites available in your system, go to:

⊙ STORES ⊙ SETTINGS ⊙ ALL STORES

Tax Class

The product tax class.

You can either use **the tax class name** (case insensitive) or **the tax class ID** of the product.

For example:

Taxable Goods

2

To know the names and the ids of the tax classes available in your system, go to:

⊙ STORES ⊙ SETTINGS ⊙ CONFIGURATION ⊙ SALES ⊙ TAX ⊙ TAX CLASSES

Visibility

The product visibility.

You can either use **the visibility name** (case insensitive) or **the numerical value associated**.

1. Not visible Individually

2. Catalog

3. Search

4. Catalog & Search

For example:

Catalog,Search

4

Status

The **status of the product**: enabled or disabled.

You can use any of the below values:

■ enabled / disabled

■ enable / disable

■ true / false

■ yes / no

■ 1 / 0

For example:

enabled

Products attributes

All attributes created in your stores and/or any attributes that are listed in another section are available in the product attributes section.

Text or textarea attribute

The text or textarea attributes accept any value without any restriction.

For example:

Your text

Yes/no attribute

The yes/no attributes can be updated with any of the below values:

- enabled / disabled
- enable / disable
- true / false
- yes / no
- 1 / 0

For example:

true

Dropdown attributes

The drop-down attributes can be updated with:

- **the option id**

For example:

4

- **the option label**

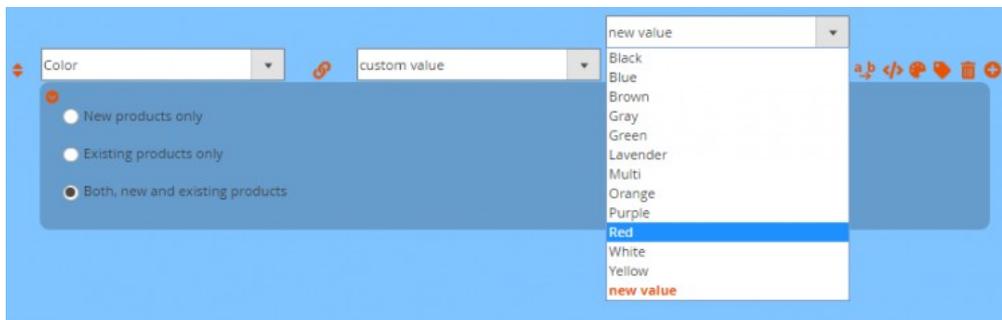
For example:

red

The values of dropdown attributes can be translated to the different store view languages as below with the **color** attribute:

red[de=rot|fr=rouge|es=rojo]

If you want to use a fixed value, you can select one from the dropdown.



Find the store codes of your store views in:

🕒 STORES 🕒 SETTINGS 🕒 ALL STORES

Multi-select attributes

The multi-select attributes can be updated either with a **list of option labels** or a **list of option ids** separated with a comma.

For example:

Men, Boys

Note that all the multi-select options are now automatically populated and you can directly select multiple values:



To know the labels and ids of the drop-down and multi-select attributes, go to:

[STORES](#) [ATTRIBUTES](#) [PRODUCT](#) [EDIT ATTRIBUTE](#) [MANAGER LABEL / OPTION](#)

Date attributes

The date attributes must be formatted with the date and time GMT:

YYYY-MM-DD hh:mm:ss

For example:

2019-06-11 15:09:52

Prices attributes

Price

The price attribute can be a **float number** or an **integer number**.

It must be formatted with a dot as decimal separator.

For example:

15,50

Special price

The special price attribute can be a **float number** or an **integer number**.

It must be formatted with a dot as decimal separator.

For example:

10,50

Special price from/to date

The special price from/to date must be formatted with the **date and the time GMT** as below:

YYYY-MM-DD hh:mm:ss

For example:

2019-06-17 11:29:51

Tier price/Group price

You have two possibilities:

- **Add Tier price/Group price**
Creates a new Tier price/Group price that is added to the existing ones.
- **Replace Tier price/Group price**
Replaces all the existing Tier prices/Group prices with the new ones.

The list of tier prices must be separated by ~ as below:

[Group id 1][Qty 1][Price 1]~[Group Id 2][Qty 2][Price 2]~...

For example:

|5|9.99~|10|7.99~...

Below is the list of the codes to use for the customer groups ids:

- *: All groups
- 0: Not logged in
- 1: General
- 2: Wholesale
- 3: Retailer
- 4: VIP Member
- 5: Private Sales Member

Weee tax

The list of taxes must be separated by ~ as below:

country1|region1|tax1~country2|region2|tax2~...

- **Country code**
It can be: FR, US, CA, NZ...
- **Region**
Use * for all states
- **Tax**
This is a fixed value

For example:

FR|*|13~UK|*|18

Stock attributes

All attributes related to the stock management in Magento are available in this section.

All these attributes must be updated with numeric values or boolean values.

Boolean values

Boolean values must be updated with one of the below values:

- 1 / 0
- yes / no
- true / false
- enable / disable
- enabled / disabled
- in stock / out of stock

For example:
in stock

Numeric values

Numeric values must be updated with one of the below types:

- **Integer number** (no white space)
- **Float number** (decimal separated by a dot)

For example:
10.50

Advanced MSI attributes

Mass Product Import & Update allows you to update all attributes related to the stock management in Advanced MSI that you can retrieve in the stocks grid from:

👉 CATALOG 🕒 STOCKS

Advanced MSI > Manage stocks

ID	Name	Type	Attrib. Set Name	SKU	Visibility	Status	Websites	Stock Status	Qty	Modal	great_lake_store Great Lake...	green_forest_store Green Fore...
From									From			
To									To			
1	Joust Duffle Bag	Simple Product	Bag	24-MB01	Catalog, Search	Enabled	Main Website	Out of Stock	78.0000	show modal	Assign source <input checked="" type="checkbox"/> Is in stock <input checked="" type="checkbox"/> Quantity <input type="text" value="58"/> Location <input type="text" value="R51 S86"/> Allow Qty below 0 <input type="text" value=""/> Use config <input type="checkbox"/> Back. d <input type="text" value="03/25/2020"/> <input type="button" value="📅"/> Backordered qty <input type="text" value="50"/> Notify Qty <input type="text" value="2"/> Use config <input type="checkbox"/>	Assign source <input checked="" type="checkbox"/> Is in stock <input checked="" type="checkbox"/> Quantity <input type="text" value="20"/> Location <input type="text" value="R25 S69"/> Allow Qty below 0 <input type="text" value=""/> Use config <input type="checkbox"/> Back. d <input type="text" value="03/12/2020"/> <input type="button" value="📅"/> Backordered qty <input type="text" value="23"/> Notify Qty <input type="text" value="5"/> Use config <input type="checkbox"/>

All these attributes must be updated with numeric values or Boolean values.

Quantity

The stock for the source.

Assign source

Is in stock

Quantity

Location

Allow Qty below 0

Use config

Back. d

Backordered qty

Notify Qty

Use config

For example:
58

Notify Qty

Notify the quantity for the source.

Assign source

Is in stock

Quantity

Location

Allow Qty below 0

Use config

Back. d

Backordered qty

Notify Qty

Use config

For example:
2

Use default for notify Qty

Use default for notify the quantity for the source.

Assign source

Is in stock

Quantity

Location

Allow Qty below 0

Use config

Back. d

Backordered qty

Notify Qty

Use config

For example:
no

Stock Status

Stock status for the source.

Assign source

Is in stock

Quantity

Location

Allow Qty below 0

Use config

Back. d

Backordered qty

Notify Qty

Use config

For example:
in stock

Location

Product location in the warehouse.

Assign source

Is in stock

Quantity

Location

Allow Qty below 0

Use config

Back. d

Backordered qty

Notify Qty

Use config

For example:
R51 S86

Backorders

Backorders status.

Assign source

Is in stock

Quantity

Location

Allow Qty below 0

Use config

Back. d

Backordered qty

Notify Qty

Use config

For example:
Allow Qty below 0

Backorders date

Backorders date in the following format:
YYYY-MM-DD hh:mm:ss

Assign source

Is in stock

Quantity

Location

Allow Qty below 0

Use config

Back. d

Backordered qty

Notify Qty

Use config

For example:
2020-03-25

Backordered qty

Backordered quantity.



The screenshot shows a form with the following fields and values:

- Assign source:
- Is in stock:
- Quantity: 58
- Location: R51 S86
- Allow Qty below 0: (dropdown menu)
- Use config:
- Back. d: 03/25/2020 (calendar icon)
- Backordered qty: 50 (highlighted in orange)
- Notify Qty: 2
- Use config:

For example:

50

Media attributes

All attributes related to the media management are available in this section.

The image/video source directory must be defined in the Image Settings

Media Gallery

The images to insert in the image gallery must be listed in the Image Gallery field as the relative path to the image source directory specified in the Image Settings tab.

The syntax to use is:

path/to/image_1.jpg

path/to/image_2.jpg, path/to/image_3.jpg

...

Each image path must be separated by one of the below characters:

- | (pipe)
- , (comma)
- ; (semi-colon)

You can use the below parameters:

- **media_type**
image (default value)
external-video
- **disabled**
Default value is 0.
- **label**
Default value is empty.
- **position**
Default value is 0.
- **destination**
Default value is empty.

For example:

```
path/to/image_1.jpg[position=1|label=image 1]
path/to/image_2.jpg[position=2|label=image 2]
path/to/image_3.jpg[destination="new_image_3.jpg"]
```

You can use additional parameters for **external-video** media type:

- **provider**
Default value is empty.
- **url**
Default value is empty.
- **title**
Default value is empty.
- **description**
Default value is empty.

For example:

```
path/to/video/preview.jpg[media_type=external-video|title=Video 1|url=https://vimeo.com/6370469]
```

In the case where you don't add the parameters in the syntax, the default values will be used.

Note that the gallery image row must stand before the base, thumbnail and small image rows because Gallery Image erases all images.

Base Image, Small Image, Thumbnail

You can associate a specific image from the Image gallery to any image attribute.

By default the image attribute available in Magento are:

- **Base Image:** image that appears on the product page
- **Small Image:** image that appears on the category page
- **Thumbnail:** image that appears on the cart, on the checkout, on the customer account...

If you have created your own image attributes in your stores, these attributes will be visible in this section.

Swatch attributes

Swatch

You can also import/update Swatch attributes using a specific syntax to import the color and the label:

```
label[color=#HEXACODE]
```

You can use the below parameters:

- **color**
Default value is empty.

In the case where you don't add the above parameter in the syntax, the default value will be used.

For example:

```
red[#FF0000]
```

```
red[color=#FF0000]
```

Category attributes

The categories associated with the products can be imported/updated with new values.

You have the possibility to choose the action between:

- Replace all categories with
- Add to the categories
- Remove from the categories

Category mapping

List your category names (case sensitive) or category ids separated with a comma:

category ID 1 [position],category ID 2 [position]...

You can use the below parameter to define the position of the products in the categories:

- **position**
Default value is empty.

For example, to import your product in position 2 in the category ID 10 and in position 1 in the category ID 13, you can write:

```
10[2],13[1]
```

```
10[position=2],13[position=1]
```

The values will be translated into category paths according to the parameters you have defined in your profile configuration from:

[⌄ ADVANCED SETTINGS](#) [⌄ CATEGORY SETTINGS](#)

In the case where you don't add the parameter in the syntax, the default value will be used.

For example, to import products in the **New Arrivals (ID: 10)** and **Dresses & Skirts (ID: 13)** categories, from the **Mapping & rules** tab of your profile configuration, map the product attribute of your choice, for example **Add to the categories** with a **custom value**.

In the following field, you can define:

```
10,13
```

PRODUCT ATTRIBUTES

Add to the categories

Apply to **The current product only**

Apply to **Both, new and existing products**

If you prefer using the category names, you could have written:

```
Default Category/Women/New Arrivals,Default Category/Women/Dresses & Skirts
```

PRODUCT ATTRIBUTES

Add to the categories

Apply to **The current product only**

Apply to **Both, new and existing products**

Merchandising attributes

The related/up-sell/cross-sell products can be imported/updated with new values.

Related / Cross sell / Up sell products

List of **related product SKU's** separated by commas.

For example:

SKU1,SKU2,SKU3...

For versions <v4.2.2, use | to separate each SKU.

Configurable product attributes

With *Mass Stock Import & Update* you can create configurable product on the fly and create relationship between children and parents.

Parent SKU

Specify the **parent SKU**.

For example:

skuparent

When updating/creating a simple product, you can add the configurable product reference so that the simple products will be automatically associated to the configurable product.

The child product must be updated/created with the configurable attributes required by the configurable product, example: color and size.

Children SKUs

List of the **children SKUs** separated by commas.

For example:

sku1,sku2,sku3

When updating/creating a configurable product, you can add the children references so that the simple products will be automatically associated to the configurable product.

Configurable attributes

Attribute code or attribute IDs separated by commas as below:

attribute code 1[position],attribute code 2[position]

You can use the below parameter to define the **position** of the configurable attributes:

■ position

Default value is empty.

For example:

size[1],color[2]...

In the case where you don't add the parameter in the syntax, the default value will be used.

When creating/updating a configurable product, the configurable attributes must be specified.

If you don't provide these attributes when creating a new configurable attributes, the product will not appear on the front-end

Downloadable product attributes

With *Mass Stock Import & Update* you can import or update downloadable products.

Links title

Title of the downloadable resource.

For example:

Downloadable file

Samples title

Title of the downloadable resource.

For example:

Downloadable sample

Link urls

Link to the file (relative or absolute path) and optional file's name separated by a pipe (|).

Link | File name

For example:

`http://www.example.com/filename.ext | My downloadable file`

Sample urls

Link to the file (relative or absolute path) and optional file's name separated by a pipe (|).

Link | File name

For example:

`http://www.example.com/filename.ext | My downloadable sample`

Grouped product attributes

With *Mass Stock Import & Update* you can create grouped products and create relationship between children and parents.

Parent SKUs

When updating/creating a grouped product, you can add the parent references so that they will be automatically associated to the grouped product.

List of related parent product SKU's separated by commas

For example:

parentsku

Children SKUs

When updating/creating a grouped product, you can add the children references so that they will be automatically associated to the grouped product.

List of related children product SKU's separated by commas

For example:

sku1,sku2...

Bundle product options

With *Mass Stock Import & Update* you can add/update or remove options from bundle products.

Add/update an option

Bundle items option title as below:

```
Bundle items option title [type=select|required=1|option_position=1
|sku=SKU ABC|selection_qty=1|is_default=0|selection_price_type=0|selection_price_value=0
|selection_can_change_qty=0|position=1]
[sku=SKU XYZ|selection_qty=2|is_default=1|selection_can_change_qty=0|position=2]
```

You can use the below option parameters:

- **type**
select / radio / checkbox / multi
Default value is **select**.
- **required**
Default value is 1.
- **option_position**
Default value is 1.

And you can use the below product parameters:

- **sku**
SKU of the option's products
It is a required value.
- **selection_qty**
Default value is 1.
- **is_default**
Default value is 0.
- **selection_price_type**
Default value is 0.
- **selection_price_value**
Default value is 0.
- **selection_can_change_qty**
Default value is 0.
- **position**
Default value is 1.

For example:

```
Primary bag [type=select|required=1|option_position=1|sku=24-
MB01|selection_qty=2|is_default=0|selection_can_change_qty=1|position=1][sku=24-
MB05|selection_qty=1|is_default=1|selection_can_change_qty=0]
```



The above example will create the following option:

Is Default	Name	SKU	Default Quantity *	User Defined
<input checked="" type="radio"/>	Wayfarer Messenger Bag	24-MB05	<input type="text" value="1"/>	<input type="checkbox"/>
<input type="radio"/>	Joust Duffle Bag	24-MB01	<input type="text" value="2"/>	<input checked="" type="checkbox"/>

In the case where you don't add the parameter in the syntax, the default value will be used.

You can add several products to an option, however each option must be on a separate line.

Remove an option

Bundle items option title to remove

For example:

Accessories



The above example will remove the "Accessories" option.

You can only remove one option per line.

Custom options

With *Mass Product Import & Update* you can import or update your product custom options.

Dropdown / Radio / Checkbox / Select / Multi-select

Custom option title and option values in the following format:

Custom option [title=title1|sku=sku1|price=10%|sort_order=1][title=title2|sku=sku2|price=10|sort_order=2]

You can use the below parameters:

■ is_require

If the custom option is a required field: **is_require=1**.

Default value is: **is_require=0**

■ title

Default value is empty.

■ sku

Default value is empty.

■ price

Note that the price can be a percentage or a fixed amount. For example:

10 or 10%

Default value is null.

■ sort_order

Default value is null.

In the case where you don't add the above parameters in the syntax, the default values will be used.

For example:

```
Engraving [is_require=1|title=Gold engraving|sku=engraving01|price=20%|sort_order=1]
```

```
[title=Silver engraving|sku=engraving02|price=10%|sort_order=2]...
```

```
Engraving [title=Gold engraving|sku=engraving01|price=50|sort_order=1]
```

```
[title=Silver engraving|sku=engraving02|price=30|sort_order=2]...
```

Text Field / Textarea

Custom option title and option values in the following format:

Custom option Text [is_require=1|title=text|sku=txt|price=10|sort_order=1|max_characters=100]

You can use the below parameters:

■ **is_require**

If the custom option is a required field: **is_require=1**.
Default value is: **is_require=0**

■ **title**

Default value is empty.

■ **sku**

Default value is empty.

■ **price**

Note that the price can be a percentage or a fixed amount. For example:
10 or 10%
Default value is null.

■ **sort_order**

Default value is null.

■ **max_characters**

Default value is null.

If no value is defined for the above parameters, the default values will be used.

For example:

Custom option Text [is_require=1|sku=txt|price=10.00|sort_order=1|max_characters=100]

File

Custom option title and option values in the following format:

Custom option Text [is_require=1|title=text|sku=txt|price=10|sort_order=1|file_extension=jpg]

You can use the below parameters:

■ **is_require**

If the custom option is a required field: **is_require=1**.
Default value is: **is_require=0**

■ **title**

Default value is empty.

■ **sku**

Default value is empty.

■ **price**

Note that the price can be a percentage or a fixed amount. For example:
10 or 10%
Default value is null.

■ **sort_order**

Default value is null.

■ **file_extension**

Default value is null.

If no value is defined for the above parameters, the default values will be used.

For example:

Custom option File [sku=upload|price=99.99|sort_order=2|file_extension=jpg,png,gif]

Replacement rules

Mass Product Import & Update allows you to control the data output by using replacement rules.

Create a replacement rule

You can create your own rule from:

- SYSTEM
- MASS PRODUCT IMPORT & UPDATE
- REPLACEMENT RULES

Click on .



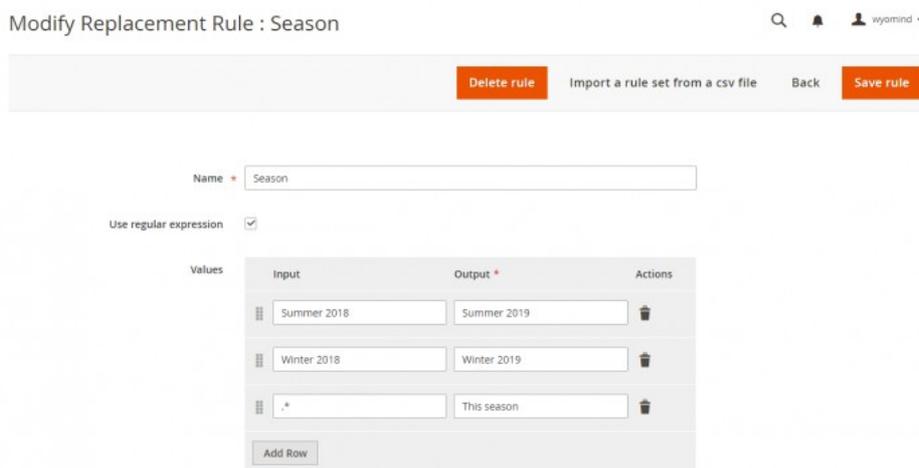
Create a replacement rule manually

You can create your own rule manually.

Fill in a name and add the values you need to replace in the **Input** column by other values in the **Output** column.

For example, the below example shows how to replace each "Summer 2018" by "Summer 2019" and all the rest with "This season".

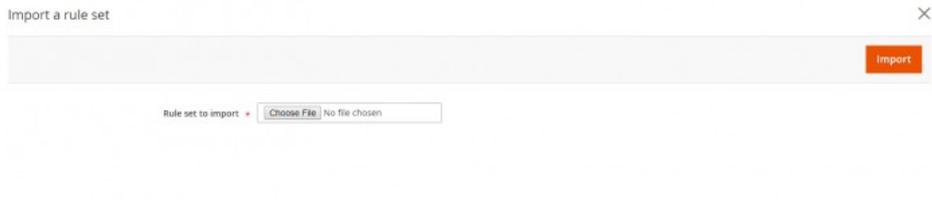
Note that you have the possibility to use regular expressions.



Import a rule set from a CSV file

You also have the possibility to import a rule set directly from a CSV file.

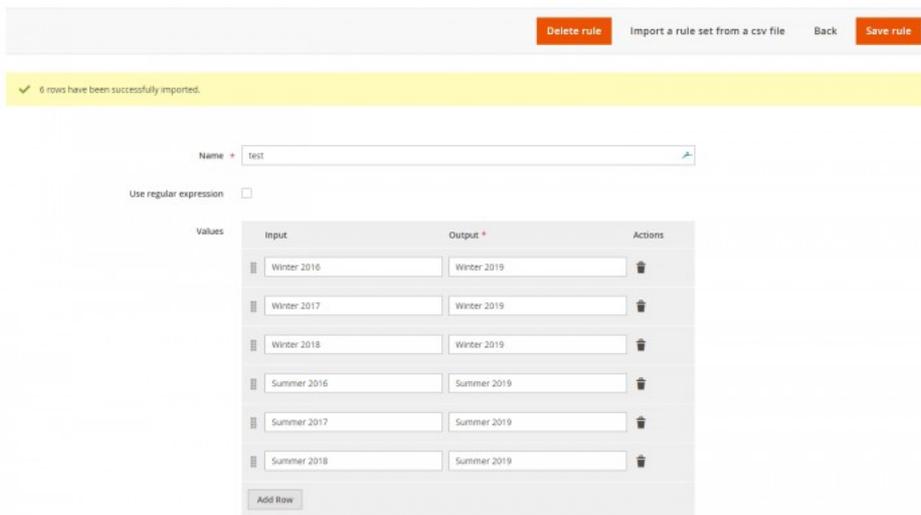
Click on and choose your CSV file. Click on .



The file must be a tab delimited CSV file. For example:

```
Winter 2016 Winter 2019
Winter 2017 Winter 2019
Winter 2018 Winter 2019
Summer 2016 Summer 2019
Summer 2017 Summer 2019
Summer 2018 Summer 2019
```

Once the lines have been imported, you receive a notification.



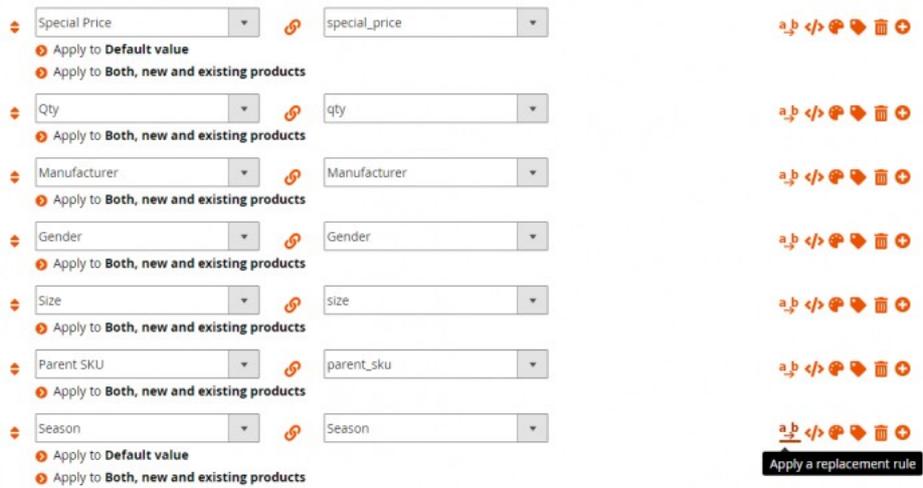
Use a replacement rule in your profile

Select a replacement rule

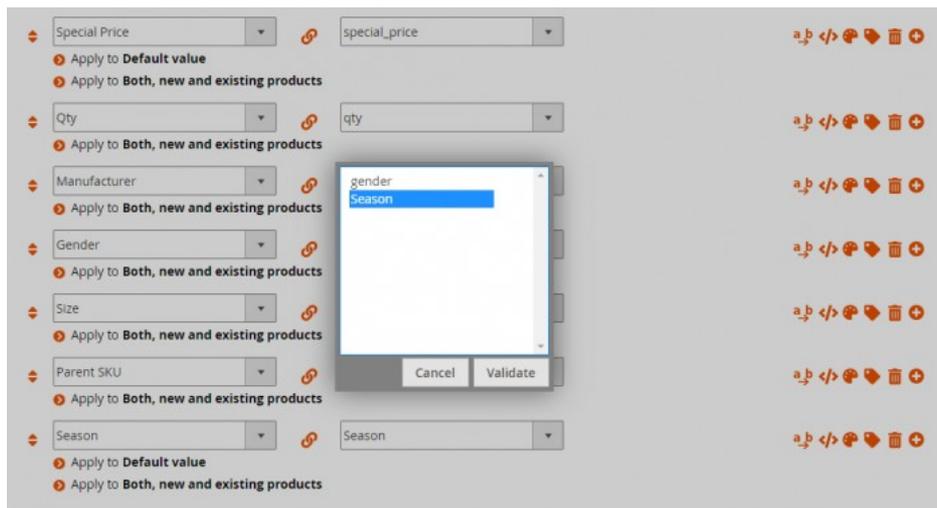
Once your rule is saved, you can use it in your profile from:

- 🕒 SYSTEM
- 🕒 MASS PRODUCT IMPORT & UPDATE
- 🕒 PROFILES

For the attribute you want to edit, click on the **ab** icon.

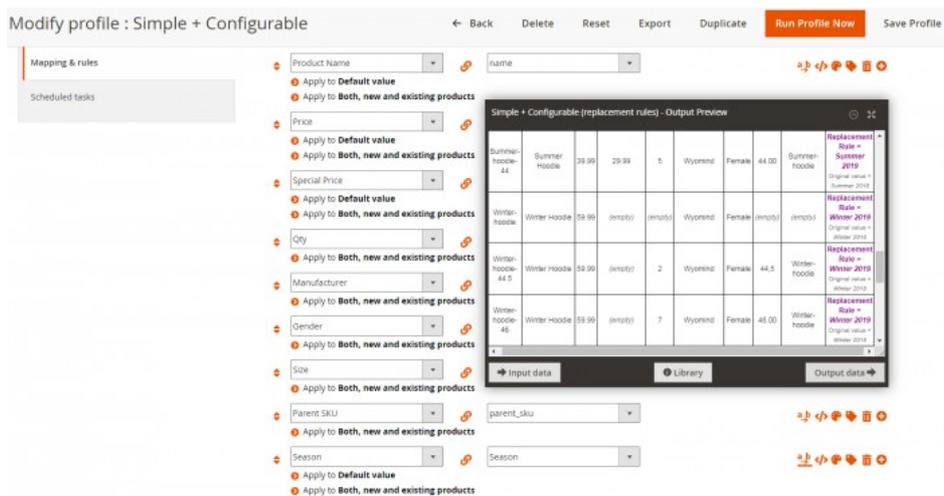


A popup displays where you can select the rule you want to apply. Select the rule and click on **Validate**.



Output the replacement rule value

If you click on **Output Data** from the blackbox, you should see the replacement rule value in purple followed by the original value in grey.



PHP API for Mass Product Import & Update

Mass Product Import & Update allows you to fully customize and control the data output by using PHP scripts in different ways.

API coding standards

For each line in the **Mapping & Rules** tab of your profile, you can add your own custom script based on a PHP syntax.

To add your custom script, click on `</>` at the end of the line you want to edit. You can then update on the file the value of each column.

Note that the `</>` button may be hidden by the blackbox.

Opening and closing tags

Every script must include the opening PHP tag `<?php`. Note that it is not necessary to use the closing tag.

```
<?php /* Your custom script */ return $self;
```

Outputting a value

return

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

\$this object

\$this->skip() or **return false (boolean false)**

To ignore the pending line (all the columns of the line).

\$this->skipRow() or **\$this->skip()** or **return false;**

To ignore the pending cell as well as all the following cells.

\$this->skipCell() or **return true;**

To ignore the pending cell.

Variable **\$cell**

You can call any column of the source file using `$cell`. It is an array that indexes each column numerically (1,2,3,...) and with a key corresponding to the name of the column header ("name","description","price"...).

\$cell["name of the column"] or **\$cell[INDEX]**

For example:

```
return $cell["description"]. " " . $cell["name"]
```

Note it is better to use associative keys. In that case, if the column order changes in the file, your scripts will still be valid.

Variable **\$self**

\$self

To retrieve the current value of the column and manipulate the data:

```
<?php return $self;
```

For example:

```
<?php return strtolower($self);
```

Examples of use

Below are some examples of use with the PHP API.

Change the separator

Transform a number with a floating value separated with a dot instead of a comma.

For example:

```
<?php return number_format(str_replace(",", ".", $self), 2, ".", ""); ?>
```

Change a text to lowercase or capital letters

Change a text in lowercase and add a capital letter as the first letter of each word.

For example:

```
<?php return ucwords(strtolower($self)); ?>
```

Retrieve images

Retrieve only one image from a list of images.

For example:

```
<?php return array_pop(explode(",", $self)); ?>
```

Group several columns

Put together several columns to create a description.

For example:

```
<?php return $self." ".$cell["product_name"].", starting at $".$cell["price"]; ?>
```

Define the visibility of a product

Define the visibility of a product depending on a column value.

For example:

```
<?php  
  
switch($self){  
    case "parent":  
        return "Catalog, Search";  
    break;  
    default:  
        return "Not visible individually";  
}  
?>
```

Map value to other ones

Map values to other ones.

For example:

```
<?php
if($self=="something"){
    return "something else";
}
else if($self=="foo"){
    return "bar";
}
else{
    return "default value";
}
?>
```

Retrieve values based on conditions

Output values based on different conditions.

For example:

```
<?php
if ($self > 200) return "0";
else return $self;
?>
```