User's manual:

PrestaShop module

Importfast

Author: Kjeld Borch Egevang

Date: 2012-04-14

Version: 4.2

E-mail: kjeld@mail4us.dk

Table of contents

1. Abstract

This document is about the *importfast* module. The module was developed step by step based on data files from a wide range of suppliers.

1.1 Features

Here is a list of the main features of the module:

- Fast import of products from CSV and XML files.
- Possible to resume the import if the maximum run time is exceeded.
- Automatically creates categories and sub-categories.
- Setup for mapping of CSV or XML fields is saved in the database.
- Support for multiple suppliers.
- Support for multiple languages.
- Images are fetched and resized on demand.
- A cronjob can handle the resizing of the images at nights.
- Optionally deactivate or keep existing products.
- Optionally delete old deactivated products.
- Optionally deactivate categories with no active products.
- Optionally deactivate products in deactivated categories.
- Optionally set quantity to zero for no longer existing products.
- Specify profit percentage based on wholesale price and retail price.
- Automatically generate HTML meta tags.

2. Introduction

2.1 History

It all started when a friend of mine wanted to import perfume via dropshipping. He had a list in XML format from a supplier. I didn't really know much of XML but I could easily see how I could convert the XML file to a CSV file so it could be used in PrestaShop.

I really like Python so I wrote a small conversion script. There were 17,000 products in the file. I soon discovered that resizing the images took around 1 second per image. It was apparent that the image conversion took way too long time so I left out the images (and really didn't know how to fix it later on). But even without the images the import was incredibly slow.

At that point I decided to write an entirely new import module. I had some experience with PrestaShop modules from a payment module I had written. This import module had a completely different scope. 1st priority was speed. 2nd priority was data file interpretation. 3rd priority was usability.

The current module is actually a bit slower than the first version I made. Back then I put some constraints on product IDs and image IDs that bought me some speed but also made it impossible to handle more than one supplier.

2.2 File formats

The module can handle two file formats: CSV and XML. Especially XML files can be infinitely complex so I will never boast the module can handle *all* XML files.

2.2.1 CSV

CSV means Comma-Separated Values. Often it's impractical to use comma (,) to separate the fields especially because decimal numbers may contain commas as well. This means that many CSV files use semicolon (;) as separator. The *importfast* module auto detects the separator. Since semicolons may occur in the descriptions it may be necessary to enclose descriptions in double quotes ("). It is also possible to insert new-lines between the double quotes.

One of the problems with CSV files is that there is no obvious way to describe complex structures like attributes. See the later section describing the attributes.

2.2.2 XML

XML means Extensible Markup Language. Apparently each supplier has their own idea about how to describe products via XML. I have been presented with a wide range of variants of XML files from suppliers. Generally, the XML

files contain category names, product IDs, names, prices etc. But some suppliers split their XML files in separate files for categories, attributes and prices.

The first problem with XML files is to identify the tag that separates the categories/products. For category files the module looks for one of: "Category", "Group". For product files the module looks for one of: "Item", "Product", "ProductInfo" and "ShopItem". If the XML file uses a completely different tag it must be selected manually.

2.2.3 Files

The *importfast* module can handle one or two files for each supplier: Either a stand-alone product file or a set consisting of a category file and a product file.

On top of this you may have a file containing only price or quantity update. This file must have its own configuration.

3. How it works

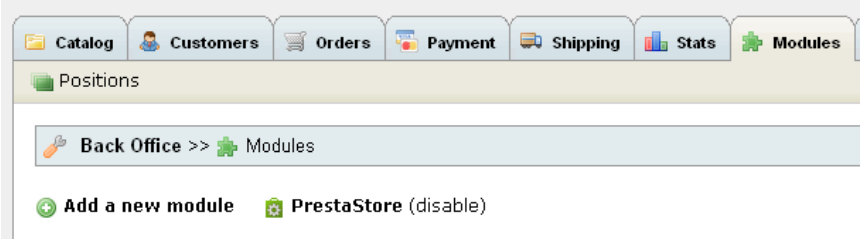
What *importfast* does is basically the same as the original import function. At least with respect to CSV files. But there are two major differences:

1. The database requests are "low level" i.e. the advanced but also time-consuming objects from the `classes` directory are not used.
2. The images are not resized immediately. If the server is low to medium speed resizing of just one image may take as long as 1 second. If your PHP max. execution time is only 30 seconds you obviously can't import many products in one go (note that there is a chance you can extend the max. execution time via `php.ini` or `.htaccess`). With *importfast* the images are resized when requested for display in the shop.

The mapping of the fields is saved in the database. If your file has many fields (especially XML files tend to have many fields) it is really nice that you don't have to set up the mapping over and over.

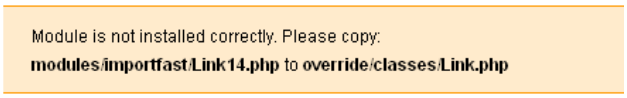
4. Installation

Install the module via the PrestaShop Modules tab.



Some servers (or actually a lot) can't handle .zip files and some installations don't have the necessary write permissions to install modules this way. If you can't install the module via the PrestaShop back office, extract it locally and copy all the files to the `modules` directory via FTP.

From the Modules tab at "importfast" click Install. After installation you may see the warning:



This means the file override/classes/Link.php could not be updated.

If you have a 1.2 version of PrestaShop:

Copy modules/importfast/Link12.php to classes/Link.php

If you have a 1.3 version of PrestaShop:

Copy modules/importfast/Link13.php to classes/Link.php

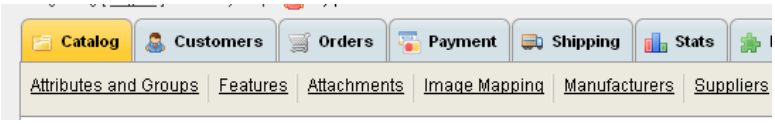
If you have a 1.4 version of PrestaShop:

Copy modules/importfast/Link14.php to override/classes/Link.php

5. Configuration

5.1 Adding a supplier

Before you can configure the *importfast* module you have to add the supplier from which you get the data file. This is done from the Catalog tab.



5.2 Import type

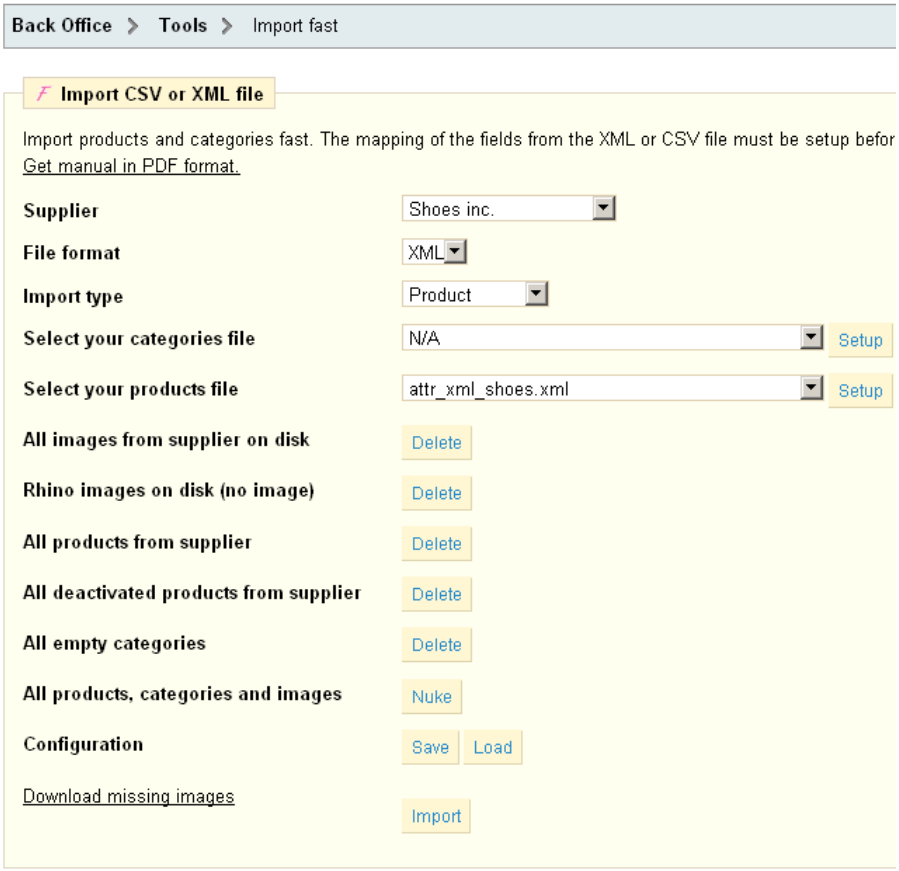
Two types of import are possible: "Product" and "Price/qty update". Select "Product" to do a full import. Select "Price/qty update" if you only want to import price and/or quantity.

Notice that for "Price/qty update" only the product table is updated. This means that if the product has attributes you can't update prices nor quantities for the variants. Instead you will have to do a full import.

5.3 Selecting file

The data files must be copied to `<admin>/import` where `<admin>` is the admin directory you are using. There is no upload function in *importfast*. This is because the data files are often so big that uploading them via "POST" wouldn't be possible anyway. So you will have to copy the files via FTP.

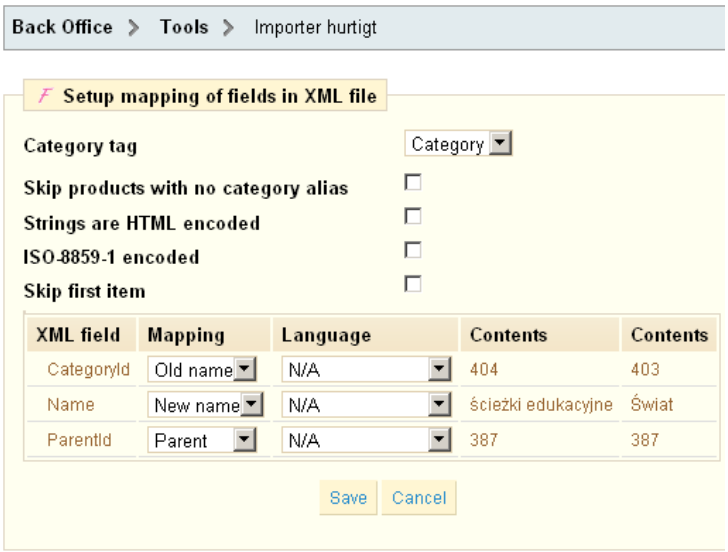
When the supplier and the files are ready you can select the file(s) you want to work with:



5.4 Category file setup: options

The categories file is in fact a simple aliases file. If the categories in your product file have the names you want, skip the next sections and go directly to "Product file setup: options" 5.7.

For the category file click Setup. The setup screen for categories looks like:



The next sections describe each of the options in the setup page.

5.4.1 Category tag [XML]

This is the XML tag that identifies a new category record. The module looks for one of: "Category", "Group". If the XML file uses a completely different tag it must be selected manually.

5.4.2 Delimiter [CSV]

The character used to delimit the fields in the CSV file.

5.4.3 Skip products with no category alias

This option allows you to filter out categories that are not mentioned in your categories files.

5.4.4 Strings are HTML encoded

Some suppliers have strings HTML encoded and use things like "&qout;". Use this option to get these strings converted to UTF-8.

5.4.5 ISO-8859-1 encoded

Some suppliers deliver the data files in ISO-8859-1. Use this option convert the contents to UTF-8.

5.4.6 Skip first item [CSV]

The first line of a CSV file often contains the field names. By checking this option the first line is ignored.

5.5 Category file setup: mapping fields

You have to map the external field names to internal field names.

The language selector is only used if your supplier has multiple languages in the data field. Otherwise just leave it at "N/A".

5.6 Category file mappings

5.6.1 N/A

Not applicable. The field is not in use.

5.6.2 Old name

The old category name. This name should be the one used in the product file.

5.6.3 New name

The new category name. This name will replace the name used in the product file. Notice that you can also specify sub-categories. Like "Men|Shoes" where "|" is the sub-category separator.

5.6.4 Parent

This indicates that the field describes a parent category.

5.7 Product file setup: options

For the product file click Setup. The setup screen has an awful lot of options:

And in the bottom of the screen you can setup HTML meta tags:

HTML meta tags	Description	Keywords	Title
Shop name	☐	☐	☐
Name	☐	☐	☐
Reference	☐	☐	☐
Supplier reference	☐	☐	☐
Manufacturer	☐	☐	☐
Category	☐	☐	☐
Features	☐	☐	☐
Short description	0	0	0
Long description	0	0	0

Probably you only need a few of the options. The next sections describe each of the options in the setup page.

5.7.1 Quantity at least

The field "Quantity at least" specifies the lowest stock number the product may have. If the quantity is lower the product is ignored. If you have checked "Deactivate no-longer existing products" an existing product will no longer be active.

5.7.2 Picking fee

Some suppliers charge a "Picking fee". I.e. every product will have this fee added.

5.7.3 Default quantity

If the data file does not specify the quantity, the "Default quantity" is used.

5.7.4 Price range for profit percent

The price ranges determine which profit percent (see below) to use. The wholesale price is compared to each of the values and the corresponding profit percentage is select.

5.7.5 Profit percent

The "profit percent" is used in the calculation of the price. The price is calculated as:

	Wholesale price not specified	**Wholesale price specified**
Price not specified	N/A	finalPrice = wholesalePrice + wholesalePrice * profitPercent / 100
Price specified	Price unchanged	finalPrice = wholeSalePrice + (price – wholesalePrice) * profitPercent / 100

5.7.6 Max. execution time

The maximal numbers of seconds the import function may run. You should set this number as high as possible. PHP only runs a limited time to avoid locking up the server. Some servers only allow PHP to run for as little as 30 seconds. You should set the "Max. execution time" slightly below. E.g. 25 seconds is a good number if the server's limit is 30 seconds. If the limit is reached you will have to resume the import.

5.7.7 Product tag [XML]

This is the XML tag that identifies a new product record. The module looks for one of: "Item", "Product", "ProductInfo" and "ShopItem". If the XML file uses a completely different tag it must be selected manually.

5.7.8 Delimiter [CSV]

The character used to delimit the fields in the CSV file.

5.7.9 Default category

The name of the category to use if category has not been setup.

5.7.10 Category separator

The product must belong to one or more categories. The "Category separator" typically has a value of "," or "|" or ">".

5.7.11 Sub-category separator

The "Sub-category separator" identifies the character (or string) the separates categories from sub-categories, sub-categories from sub-sub-categories etc. Typical values are "," or "|" or ">".

5.7.12 Category leader to ignore

In some XML files the category name has an unwanted prefix. You can get rid of this prefix by specifying it as "Category leader to ignore". This option is very rarely used.

5.7.13 Image separator

Like multiple categories needs a separator so does multiple images. This is just one way multiple images can be expressed.

5.7.14 URL prefix

The images in the data file are quite may only consist of the base name of the file. In this case you can set up the "URL prefix" that must be prepended in order to get the full valid URL. Examples could be:
`http://www.yoursupplier.com/images/medium/` or

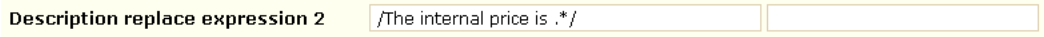
`/home/monty/prestashop/images/medium/` where the latter expects that image files can be accessed locally.

5.7.15 Description replace expression 1

This regular expression allows you to substitute texts in the short and/or long description. Regular expressions are very flexible but also a bit complicated. Here is a simple example where a reference to a supplier's server should be replaced with a local server:

5.7.16 Description replace expression 2

One more regular expression for the description fields. Here is an example that eliminates the text starting with "The internal price is" and to the end of the description:



5.7.17 Default tax

The tax may be specified in each product record. If you just want to use a default tax for all products you can select it here.

5.7.18 Delete products deactivated for more days than

Normally it's a bad idea to delete products. You need old products (though they may be set not active) if you get customer returns and want to know what products were actually in the order. So, generally you should *never* delete a product, even if your supplier no longer has it on the list.

But in some cases the supplier changes the product line so often that your product database is growing unreasonably large. In this case it may be a good idea to age out products that are older than a specified number of days. This is what you use this option for.

5.7.19 Deactivate no-longer existing products

You should check this option if you want to make sure that products that are *not* in your suppliers' list anymore must be set to not active in your shop.

5.7.20 Deactivate free products

Some date files have products where the price for unknown reasons is set to zero. Check this option to avoid getting free products in your shop by mistake.

5.7.21 Deactivate zero weight products

Some date files have products where the weight for unknown reasons is set to zero. Check this option to avoid getting weightless products in your shop by mistake.

5.7.22 Deactivate categories with no active products

If a category has no active products it may confuse your customer. This option deactivates the categories as well.

5.7.23 Deactivate products in deactivated categories

If you want to deactivate a category (with all its sub-categories), maybe because you don't want any of the contained products in your shop, you can use this option. This way irrelevant products are automatically deactivated.

5.7.24 Set quantity to zero for no-longer existing products

Products that are not listed in the data will get the quantity set to zero.

5.7.25 Ignore products with no picture

Use this option if you demand pictures for all your products.

5.7.26 Only import new products

Existing products for the selected supplier will not be updated.

5.7.27 Comma as decimal separator

Most data files use full stop (dot) as decimal separator. If your data file use comma (as e.g. Germany uses for numbers) you should check this option.

5.7.28 Products in category leaves

By default all a product are put in the category or sub-category specified. But the product is also put in all the parent categories. You can use this option if you only want the products to be put in the leaves of the category tree.

5.7.29 Products remain in categories

If you don't really care about the category structure suggested by your supplier, you can use this option to make sure the products stay in the categories you have manually selected. You probably want to use this option in combination with "Deactivate products in deactivated categories" [5.7.23].

5.7.30 Strings are HTML encoded

Some suppliers have strings HTML encoded and use things like "&qout;". Use this option to get these strings converted to UTF-8.

5.7.31 ISO-8859-1 encoded

Some suppliers deliver the data files in ISO-8859-1. Use this option convert the contents to UTF-8.

5.7.32 Skip first item [CSV]

The first line of a CSV file often contains the field names. By checking this option the first line is ignored.

5.8 Product file setup: mapping fields

You have to map the external field names to internal field names. For this there is a primary and a secondary mapping. In most cases only the primary mapping is used.

The language selector is only used if your supplier has multiple languages in the data field. Otherwise just leave it at "N/A".

5.9 Product file primary mappings

5.9.1 N/A

Not applicable. The field is not in use.

5.9.2 Valid

The value of the field must indicate whether the product is valid. Otherwise it is completely ignored. "Valid" in this context means a value equal to the value you specify in the field just below. Example:

5.9.3 Active

The value of the field must indicate whether the product is active. Otherwise the product is added/updated but not set active. "Active" in this context means a value equal to the value you specify in the field just below. Example:

5.9.4 Supplier reference

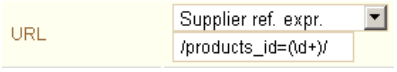
The supplier reference is the text or number that uniquely identifies the product. In the database there is room for 32 characters. If the suppliers' reference is a URL you probably want to use "Supplier ref. expr." [5.9.5].

The supplier reference in connection with the supplier ID uniquely defines the product to the *importfast* module.

If you have several products with the same supplier reference they are counted as duplicate products. If the attributes are set up properly, products with the same supplier reference are treated as variants.

5.9.5 Supplier ref. expr.

The suppliers' reference may be a URL or similar. Since the supplier reference field in the database is limited to 32 characters you can use this setting to extract only the ID part of string. This is done with a regular expression that you write in the field just below. An example could be:



where the ID just after "products_id" is extracted.

5.9.6 Quantity

The number of items in stock.

5.9.7 Name

The product name.

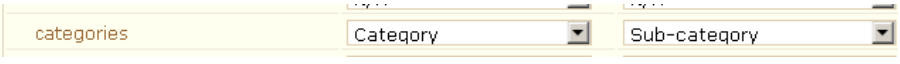
5.9.8 Category

The name of "root" category. If the product belongs to more than one category the categories must be separated by "Category separator" [5.7.10].

For XML files, the category tree may also be written like:

```
<categories>
  <tree>
    <entry>Audio</entry>
    <entry>Cleaning</entry>
    <entry>Maintenance</entry>
  </tree>
  <tree>
    <entry>Computer</entry>
    <entry>Parts</entry>
    <entry>Components</entry>
  </tree>
</categories>
```

resulting in two category branches: Audio->Cleaning->Maintenance and Computer->Parts->Components. In this case the mapping must look like:



to indicate that the category name as well as the sub-category name can be found via the XML tag "categories".

In some XML files the categories enclose several products like:

```
<STORE>
  <LIST>
    <CATEGORY name='Clothes > Bare Essentials'>
      <PRODUCT ITEM='1969'>
       price, name, description etc.
      </PRODUCT>
      <PRODUCT ITEM='2011'>
       price, name, description etc.
      </PRODUCT>
    </CATEGORY>
  </LIST>
<STORE>
```

(notice the sub-category separator is ">" in this example). The setup looks like:

Notice that only the inner tag "name" is shown. If you position the mouse cursor over "name" you will get a tooltip with the full path: "STORE|LIST|CATEGORY|name".

5.9.9 Category ID

The data file may contain category IDs instead of category names. The category ID may refer to root category or a sub-category (or sub-sub etc.). Notice that typically this is *not* what you want. Instead you should consider using a categories file.

5.9.10 Sub-category

The name of the sub-category. Primary mapping of sub-category occurs in some CSV files.

5.9.11 Sub-sub-category

The name of the sub-sub-category. The primary mapping of sub-sub-category occurs in some CSV files.

5.9.12 Sub-sub-sub-category

The name of the sub-sub-sub-category. The primary mapping of sub-sub-sub-category occurs in some CSV files.

5.9.13 MOQ

Minimum order quantity. You probably don't need this. Anyway, the wholesale price, the weight and the reduction (specific reduction) are multiplied by "MOQ".

5.9.14 Price

This is the price of the product. If you want to add profit to the price you should use wholesale price instead. Refer to "Profit percent" [5.7.5] to learn how the final price can be calculated based upon wholesale price and price.

5.9.15 Wholesale price

The wholesale price. Refer to "Profit percent" [5.7.5] to learn how the final price can be calculated based upon wholesale price.

5.9.16 On sale

If the value is not "0" or "N" the product is marked as on sale.

5.9.17 Tax rate

The tax rate for the product. The tax rate is matched against any of the already defined tax rates.

5.9.18 Reduction price (PS v1.3)

The amount (incl. tax) the price of the product is reduced when the time is in the interval "Reduction from" and "Reduction to".

5.9.19 Reduction percent (PS v1.3)

The percentage the price of the product is reduced when the time is in the interval "Reduction from" and "Reduction to".

5.9.20 Reduction from (PS v1.3)

Start date (and time) for reduction.

5.9.21 Reduction to (PS v1.3)

End date (and time) for reduction.

5.9.22 Reduction amount (PS v1.3)

The amount the wholesale price should take when the time is in the interval "Reduction from" and "Reduction to". The amount is adjusted with "Profit percent".

5.9.23 Specific price (PS v1.4)

The amount the wholesale price should take when the time is in the interval "Specific reduction from" and "Specific reduction to". The amount is adjusted with "Profit percent".

5.9.24 Specific reduction (PS v1.4)

The amount (incl. tax) the price of the product is reduced when the time is in the interval "Specific reduction from" and "Specific reduction to".

5.9.25 Specific quantity (PS v1.4)

The least quantity for which the reduction is valid.

5.9.26 Specific reduction type (PS v1.4)

The reduction type. Possible values are "amount" and "percentage". Default is "amount".

5.9.27 Specific reduction value (PS v1.4)

The value used for the reduction ("amount" or "percentage").

5.9.28 Specific reduction from (PS v1.4)

Start date (and time) for reduction.

5.9.29 Specific reduction to (PS v1.4)

End date (and time) for reduction.

5.9.30 Additional shipping cost (PS v1.4)

Additional shipping cost for the product.

5.9.31 Online only

Product is online only.

5.9.32 Upc

Universal product code.

5.9.33 Minimal quantity

The minimal quantity that must be purchased per sale.

5.9.34 Unity

Unity for the product.

5.9.35 Unit price ratio

Price per unit.

5.9.36 Condition

Condition of the product.

5.9.37 Width

Width of the product.

5.9.38 Height

Height of the product.

5.9.39 Depth

Depth of the product.

5.9.40 Reference

The reference shown in the shop.

5.9.41 Manufacturer

The name of the manufacturer of the product.

5.9.42 EAN13

The European Article Number (13 digit barcode).

5.9.43 Eco tax

The Ecological taxation.

5.9.44 Eco tax +VAT

The Ecological taxation without VAT. VAT (tax) is automatically added.

5.9.45 Weight

The weight of the product. You can specify the weight unit you prefer via Preferences->Localization.

5.9.46 Short description

The short description of the product. If this field is longer than 400 characters you cannot update the product description manually. In PrestaShop 1.4 you can adjust the max. length in the back office.

5.9.47 Long description

The long description (known as "description" in the database) of the product.

5.9.48 Short or long description

If the "short description" is empty the text goes into "short description" and "long description" (known as "description" in the database) is left empty. Otherwise the text goes into "long description".

5.9.49 URL

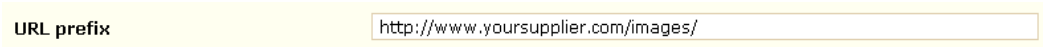
The Uniform Resource Locator identifies the image. Typically URLs start with "http://" but you may also refer to a local file in the file system. There may be multiple images separated by "Image separator" [5.7.13].

In XML files multiple images can be expressed as:

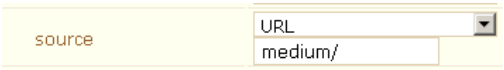
```
<images>
 <image>img1.jpg</image>
 <image>img2.jpg</image>
</images>
```

or similar.

In the field just below you can specify text you want to append to the "URL prefix" [5.7.14]. Example:

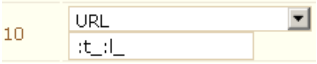


and



will cause to module to look for images in `http://www.yoursupplier.com/images/medium/`.

It is also possible to substitute part of the image path with ":". Example:



will substitute all occurrences of "t_" with "l_".

5.9.50 URL1

One more image.

5.9.51 URL2

And yet another image.

5.9.52 URL3

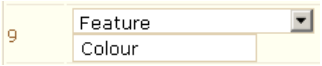
And yet another image.

5.9.53 URL4

And yet another image.

5.9.54 Feature

You can specify features for a product. The feature name is put in the field just below. Example:



5.9.55 Attribute group

An attribute group consists of a combination of attribute names and attribute values. The resulting combination may lead to an impact on price [5.9.61] and/or weight [5.9.62] and specify a quantity [5.9.60].

5.9.56 Attribute group comb.

In some case (typically with XML files) there are a number of attribute names and values but without information about quantity. This is sort of a wildcard way to use attributes. By choosing "Attribute group comb." all possible combinations are generated automatically. Example:

```
<Attributes Name="Color">
  <AttributeValues Title="Black" PriceAdjust="0.00">
  </AttributeValues>
  <AttributeValues Title="Red" PriceAdjust="0.10">
  </AttributeValues>
</Attributes>
<Attributes Name="Size">
  <AttributeValues Title="Small" PriceAdjust="-0.20">
  </AttributeValues>
  <AttributeValues Title="Medium" PriceAdjust="0.00">
  </AttributeValues>
  <AttributeValues Title="Large" PriceAdjust="0.50">
  </AttributeValues>
</Attributes>
```

Will result in all six combinations:

1. (Black, Small) and priceAdjust = -0.20
2. (Black, Medium) and priceAdjust = 0.00
3. (Black, Large) and priceAdjust = 0.50
4. (Red, Small) and priceAdjust = -0.10
5. (Red, Medium) and priceAdjust = 0.10
6. (Red, Large) and priceAdjust = 0.60

5.9.57 Attribute name

The attribute name. Can be used in connection with "attribute value" if there are no attribute groups.

5.9.58 Attribute value

The attribute value.

5.9.59 Attribute value +adj.

This is a specialized option aimed at XML files where the price adjustment is separated from the value with "|" like:

```
<option>
  <option_name>Special Offer</option_name>
  <option_value>No Thanks|+|0.00</option_value>
  <option_value>Extra batteries|+|4.99</option_value>
</option>
<option>
  <option_name>Special Offer 2</option_name>
  <option_value>No Thanks|+|0.00</option_value>
  <option_value>Extra connector |+|2.50</option_value>
</option>
```

5.9.60 Attribute quantity

The quantity for the attribute combination.

5.9.61 Attribute price adjust

The price adjustment (positive or negative number) for the attribute combination.

5.9.62 Attribute weight adjust

The weight adjustment (positive or negative number) for the attribute combination.

5.9.63 Attribute reference

The reference associated with the attribute combination.

5.9.64 Attribute supplier ref.

The supplier reference associated with the attribute combination.

5.9.65 Attribute group dash

This is a specialized option aimed at XML files where the attribute name and value are written as:

```
<option_type>Size-Size</option_type>
<option_value>XL-XL</option_value>
```

The last of the two texts separated by "-" is used. "Attribute group dash" shall select the attribute name.

5.9.66 Attribute value dash

This is a specialized option aimed at XML files as described in [5.9.63]. "Attribute value dash" shall select the attribute value.

5.9.67 Attribute default

Indicates the default combination for the product.

5.9.68 Attribute enabled [XML]

For XML files certain attributes may be listed but not active. The field may have the value "enabled" or "disabled". If the attribute value is disabled it is skipped.

5.9.69 Property

You probably don't need this setting. It is aimed at one particular XML format that has the tag "properties" and numbered identifiers pointing out what the property is. Contact me for further details if you have an XML file with tags like "productList", "internetText" and "properties".

5.9.70 Available now

The text to use if the product is available now.

5.9.71 Available later

The text to use if the product is available later.

5.9.72 Location

The location of the product.

5.9.73 Accessories

Accessories may be specified with the supplier reference for another product. Multiple accessories must be separated by "|". Example: "dev_1181|dev_1182".

If the accessory is not a product from this supplier you can specify the supplier name followed by a ":". Example: "Shoes inc.:dev_1181".

5.9.74 Customer field

If you have modified that database and added a customer field to the product table, you can use this setting to point out your customer field. The field name is written in the field just below.

5.9.75 Discount qty/pct [CSV]

For CSV files it is possible to specify a quantity discount. Use ":" as separator. E.g. "5:10" means a quantity of 5 gives 10 percent discount. A list of discounts can be specified by separating the qty/pct sets by ",".

5.9.76 Meta description

The meta description for the product.

5.9.77 Meta keywords

The meta keywords for the product.

5.9.78 Meta title

The meta title for the product.

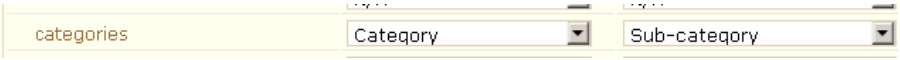
5.10 Product file secondary mappings

5.10.1 N/A

Not applicable. The field is not in use.

5.10.2 Sub-category

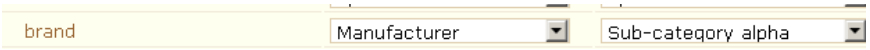
The field is also used as sub-category. Example:



Quite often this is the case.

5.10.3 Sub-category alpha

The result of some XML files may lead to very large sub-categories. A way of handling this could be to use the manufacturer name to sub divide the sub-category. You can do this like:



5.10.4 Description extra

A specialized setting where a tailing table (HTML tag "table") is extracted and put in the long description (known as "description" in the database).

5.10.5 Append to short description

Append the text to the short description.

5.10.6 Append to long description

Append the text to the long description.

5.10.7 Add to price

Add the (numeric) contents of the field to the price.

5.10.8 Reference

Also use this field for as reference.

5.10.9 Image has this ID (.jpg)

The image file can be expressed as this field appended with ".jpg".

5.10.10 Image has this ID (.gif)

The image file can be expressed as this field appended with ".gif".

5.10.11 Image has this ID

The image file is identical to the value of this field.

5.10.12 Manufacturer

The name of the manufacturer of the product.

5.10.13 On sale

If the value is not "0" or "N" the product is marked as on sale.

5.11 Language mappings

If the data file has multiple languages, select the proper language.

5.12 HTML meta tags

If you don't have meta description, meta keywords or meta title in your data file you can generate them automatically. The setup looks like:

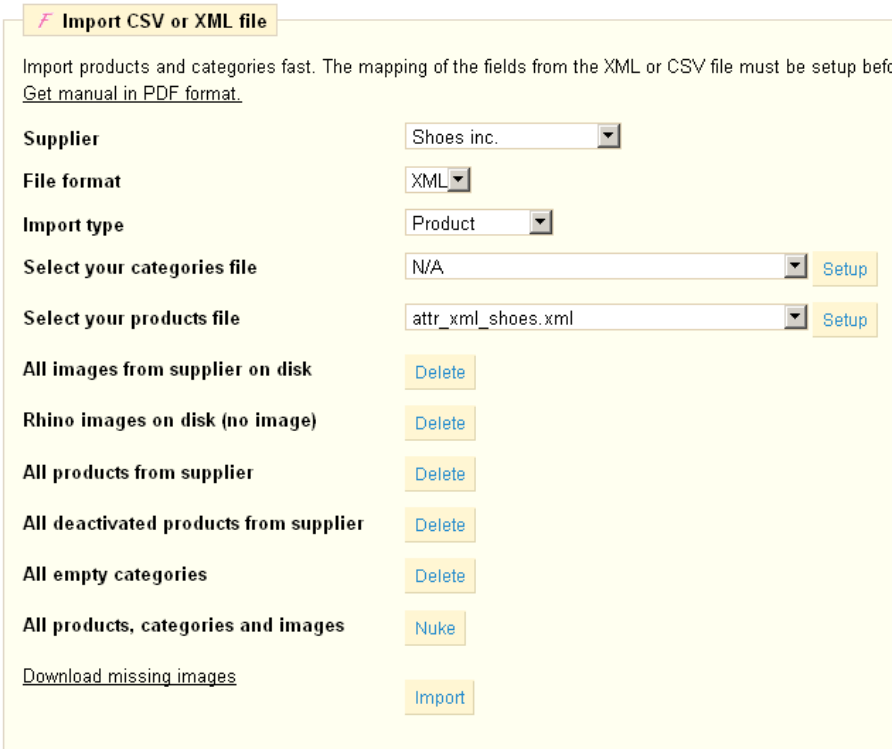
HTML meta tags	Description	Keywords	Title
Shop name	☐	☐	☐
Name	☐	☐	☐
Reference	☐	☐	☐
Supplier reference	☐	☐	☐
Manufacturer	☐	☐	☐
Category	☐	☐	☐
Features	☐	☐	☐
Short description	0	0	0
Long description	0	0	0

You check the boxes where you want data to go into the meta tags. For the descriptions you specify the max. length of the text that should go to the meta tags. Only 55 characters are shown in Google and only 255 characters

are indexed, so for performance reasons there is no need to specify too long descriptions.

6. Delete buttons

On the main page you find four Delete buttons and one Nuke button.



The delete buttons should be handled with care. Back up your files and your database before you experiment.

6.1 All images from supplier on disk [Delete]

If you are in doubt whether all your images from the selected supplier are in sync you can delete all the associated images in `images/p`. Be careful: As this will remove all the images from the disk so you will have to copy and resize them again.

6.2 Rhino images on disk (no image)

If the image could not be found the default image is a cute rhino (indicating no image). If you want to retry fetching the missing images you can delete all the rhino images. You can replace the file noimage.jpg if you want another "no image" image.

6.3 All products from supplier [Delete]

If you have changed your setup drastically (e.g. selected a different field as supplier reference) you may have to delete all the products from the selected supplier.

6.4 All deactivated products from supplier [Delete]

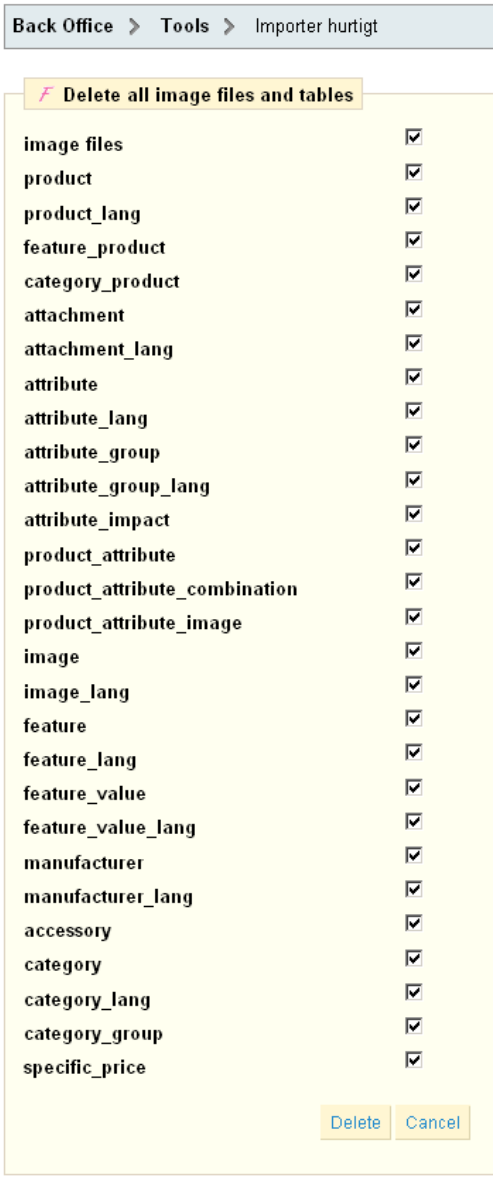
Generally old products should just be deactivated so you don't loose track. If, for some reason, you want to cleanup all the products you can use this button.

6.5 All empty categories [Delete]

Some categories may be hanging around with no products. Use this button to remove all the empty categories.

6.6 All products, categories and images [Nuke]

The Nuke button leads you to another page:



where you can select which tables you want to delete. Be careful Nuke really cleans up thoroughly.

7. Configuration management

The configuration for the selected supplier may be saved and loaded.

7.1.1 Configuration Save

Save configuration in XML format.

7.1.2 Configuration Load

Load configuration in XML format.

8. Cronjobs

8.1 The script imagecron.php

`imagecron.php` resizes as many images as possible within 30 seconds. You may discover that it takes as long as 1 second per image.

You can find a link to the `imagecron.php` script on the main page of the import module:

You can use the link in a cronjob if you want to be sure all your images have been resized. By default the images are resized when requested for display in the shop. The drawback of this strategy is that load time is significantly higher the first time a product (and the related products on the page) is displayed.

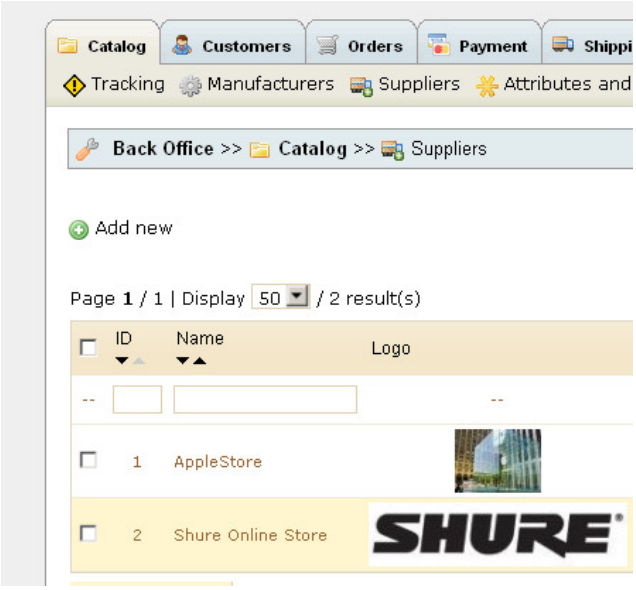
Notice that Google and other search engines will eventually cause all your images to be resized.

I can recommend http://www.cronjob.de for the cronjob (only in German but easily understandable).

I can also recommend http://www.onlinecronjobs.com.

8.2 The script importcron.php

`importcron.php` is an example script. If you want to use it, it should be placed in your `<admin>` directory. You will have to know a little about PHP in order to adjust it to suit your needs. You should set `$productFileName` to the file name you use for your data file. `$confSupplierId` and `$supplierId` should be set to the same value namely the numeric ID of your supplier. You can find the numeric ID on the supplier list:



For administrative reasons you may want to have several suppliers using the same *importfast* configuration. This can be achieved by adjusting `$supplierId` and leaving `$confSupplierId` with the ID of a common configuration.

If you import a lot of products you may be required to resume the import. You do this by appending "`?resume=1`" to the URL the subsequent times you call the `importcron.php` script.

8.3 Re-indexing search function

When you have imported your data file you must remember to re-index the search function. Go to the tab Preferences->Search and find the link "Add missing products to index". Click the link to re-index. If you activate your import from a cronjob you should add the re-index as a job just after the import.

9. Performance

Even though *importfast* is fast you should be aware of some of the issues that may slow it down.

9.1 Multiple languages

If you have enabled multiple languages you should disable all the languages you don't use. The import time is increased almost linearly with respect to the number of languages you have enabled.

9.2 Attributes

Attributes may require a lot of processing. There is not much to do about it, but don't be surprised if the import time is longer for a data file with attributes than for a data file without.

9.3 Second import

The first time you import into a clean database is radically faster than the second time. This is because has to keep track of all the existing products, categories etc.

9.4 Memory consumption

The memory consumption is a bit high so on some servers you may experience out of memory errors. If you are lucky you can just uncomment the line:

```
ini_set('memory_limit', '64M');
```

Or perhaps you can set it even higher. On some servers you can raise the limit in `php.ini` or in `.htaccess`.

If you are unlucky you must contact you host provider and try to get a higher memory limit.

10. Examples

10.1 CSV

Here is an example of a CSV file with attributes. The separator is ";". Actually there are only two products but the first product comes in 6 variants.

```
ID;Active;Name;Categories;Manufacturer;
Description;Quantity;Price;ImageURLs;Options
1014;1;Fancy dress;Women's clothes;Elle;
Very nice dress;5;100.25;img1.jpg;Color:Red,Size:Small
1014;1;Fancy dress;Women's clothes;Elle;
Very nice dress;4;100.05;img1.jpg;Color:Blue,Size:Small
1014;1;Fancy dress;Women's clothes;Elle;
Very nice dress;0;100.15;img1.jpg;Color:Red,Size:Medium
1014;1;Fancy dress;Women's clothes;Elle;
Very nice dress;5;100.25;img1.jpg;Color:Blue,Size:Medium
1014;1;Fancy dress;Women's clothes;Elle;
Very nice dress;3;100.25;img1.jpg;Color:Red,Size:Large
1014;1;Fancy dress;Women's clothes;Elle;
Very nice dress;5;100.25;img1.jpg;Color:Blue,Size:Large
1015;1;Non-fancy dress;Women's clothes;Elle's;
Very nice dress;5;100.25;img1.jpg;
```

10.2 XML

Here is an example of an XML file with attributes.

```
<?xml version="1.0" encoding="UTF-8"?>
<products>
  <item>
    <id>1181</id>
    <name>Modern leather shoe</name>
    <model sku="dev_1181">1181-dev_1181</model>
    <category>Men|Shoes,Leather|Shoes</category>
    <manufacturer>Semprini</manufacturer>
    <description>
      A great shoe in real leather. Made for everyday use.
    </description>
    <price>2.9900</price>
    <msrp>6.0000</msrp>
    <quantity>9</quantity>
    <weight>1.00</weight>
    <image>big/leathershoe.jpg</image>
    <attributeGroup>
      <attribute name="Shoe size">
        <value>1 - 33</value>
      </attribute>
      <attribute name="Color">
        <value>White</value>
      </attribute>
      <quantity>4</quantity>
      <priceAdjust>+0</priceAdjust>
      <weightAdjust>+0</weightAdjust>
      <ref>variant1</ref>
    </attributeGroup>
    <attributeGroup>
      <attribute name="Shoe size">
        <value>1 - 33</value>
      </attribute>
      <attribute name="Color">
        <value>Red</value>
      </attribute>
      <quantity>4</quantity>
      <priceAdjust>+0</priceAdjust>
      <weightAdjust>+0</weightAdjust>
      <ref>variant2</ref>
    </attributeGroup>
    <attributeGroup>
      <attribute name="Shoe size">
        <value>1 - 33</value>
      </attribute>
      <attribute name="Color">
        <value>Black</value>
      </attribute>
      <quantity>19</quantity>
      <priceAdjust>+0</priceAdjust>
      <weightAdjust>+0</weightAdjust>
      <ref>variant3</ref>
    </attributeGroup>
    <attributeGroup>
```

```
      <attribute name="Shoe size">
        <value>2 - 34</value>
      </attribute>
      <attribute name="Color">
        <value>Red</value>
      </attribute>
      <quantity>7</quantity>
      <priceAdjust>+1</priceAdjust>
      <weightAdjust>+1</weightAdjust>
      <ref>variant4</ref>
    </attributeGroup>
  </item>
</products>
```