Product/Price Crawler

Product/Price crawler is used to populate both products and prices on your website at the same time. It is an improvement over the two old separate Product and Price crawlers. So that instead of setting up two crawlers separately for a single merchant website, you can setup a single crawler that will import both products and prices.

As the old crawlers worked, it is separately configured for each website. It means that it will crawl only a single website for which it is configured in order to fetch products, images, description and prices.

You can add multiple configurations to configure separate websites at a time. Each configuration is good for a single website.

You will have to enter set of configuration parameters properly for successful crawling. In this tutorial we will learn how to successfully configure and crawl for a website.

Suppose you want to fetch products from www.bizrate.com then you will set parameters of the crawler based on the html tags of this website (you need to know basic html to set parameters). These settings will be working for only this website and will fetch products from this website. Once you are done with this website, you can reconfigure for any other website. Usually one or two websites are enough to extract products.

Some Common Questions:

*** How each product will be inserted into a particular category?**
While setting parameters for the crawler you will also specify the category extraction start and end parameters (we will discuss this in detail), which will decide the category in which to insert extracted products.

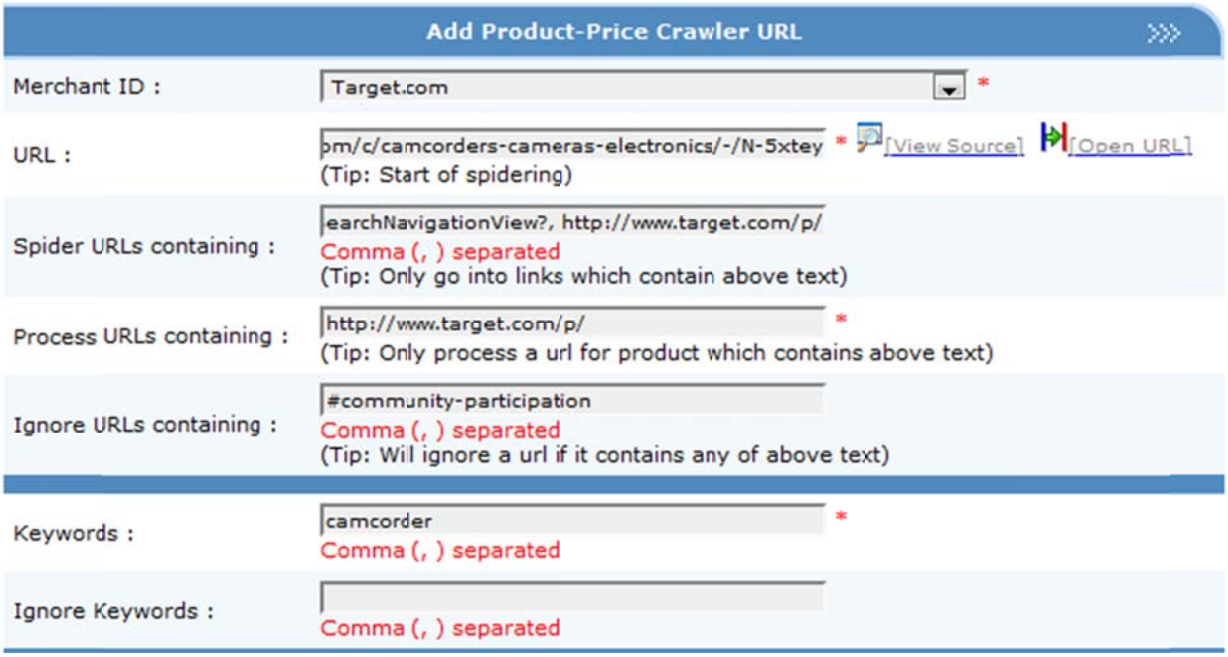
*** How does the script know which merchant this product's price belongs to?**
This is pretty simple, when setting up a crawler; you will have a dropdown of merchants to select one, this is where the script is instructed about the owner of inserted prices.

*** The pages crawled will have products other than we want to insert, how can we prevent them from getting inserted into our database?**
Again at the time of specifying parameters for crawler; you will also specify set of keywords and ignore-keywords. The crawler reads the chunk of webpage and matches the information extracted with the keywords and ignore-keywords. In order to add product and price extracted into the system, it must match the keywords and must not match ignore-keywords.

Manage Product/Price Crawlers

To explain how it works, we will create a crawler for the website target.com and import its camcorder category products.

Go to Content > Categories > Product/Price Crawler > Manage Product/Price Crawler. Now hit the Add button to add a crawler and you will see a large form appear on the screen. We will discuss all parts of that form one by one:



Merchant ID: As we stated earlier in this document, you will have to select a merchant from the dropdown list to tell for whom this crawler is created. You must have the merchant already existing in your database. Here we are using Target.com for our example.

URL: This is the starting url for the crawler. We are using www.target.com/c/camcorders-cameras-electronics/-/N-5xtey as the starting url. This is the url for camcorder categories on target.com

Spider URLs containing: It should contain a phrase which should be present in the subsequent urls to be spidered. Meaning, when you will go to the above url, you will notice there are links to the other categories and products on that page as well as links to about us and privacy policy. You dont want your crawler to waste time in unnecessary pages so you instruct it to spider only those urls which contains that particular phrase.

You can provide more than one phrases separated by commas.

We are using these phrases here: http://www.target.com/SearchNavigationView?, http://www.target.com/p/
The first one being used to display next pages of camcorder products listing while the second one is used to go to the actual product pages.

Process URLs Containing: This should contain a phrase that is present in the product detail page urls, so that only those pages are processed for product extraction.

In our example, we use http://www.target.com/p/ so that the crawler will only process these urls.

Note: Process URLs phrase should be the sub-set of Spider URLs phrase. It must contain one (or more) entries from Spider URLs only. In other words, a page spidered for URLs can only be processed for product extraction BUT it is not necessary that all pages spidered are processed.

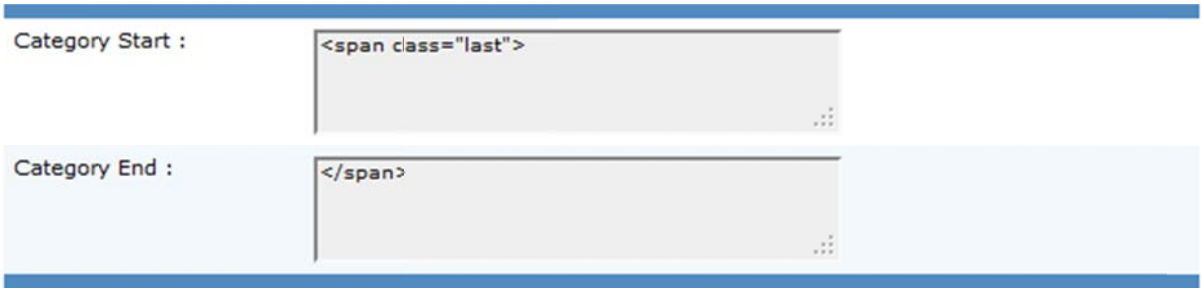
Ignore URLs Containing: This could have a phrase that you want to ignore for spidering.

In our example, the listing pages on target.com have product detail page urls starting with http://www.target.com/p/ but also the product reviews page urls too start with http://www.target.com/p/, so to ignore those review pages, we use this field with a value *#community-participation* which is only contained in the review page urls and not in the product page urls.

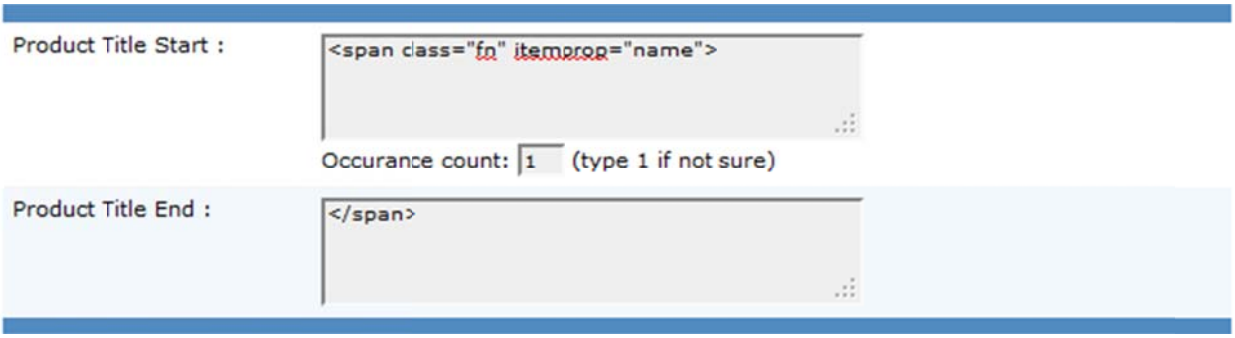
Keywords: You can provide keywords here separated by commas such that if these keywords exist in the product page content, only then that products is extracted for addition.

Ignore Keywords: You can provide ignore keywords here separated by commas such that if these keywords exist in the product page content, then those products are NOT extracted for addition.

Chunk Start and Chunk End: A chunk is the block of html tags/codes in which a product is present with its complete information. To extract the chunk, we need to tell the crawler the starting and ending tags/html of the chunk. This start and end chunk should be unique so that the crawler can easily identify the chunk.



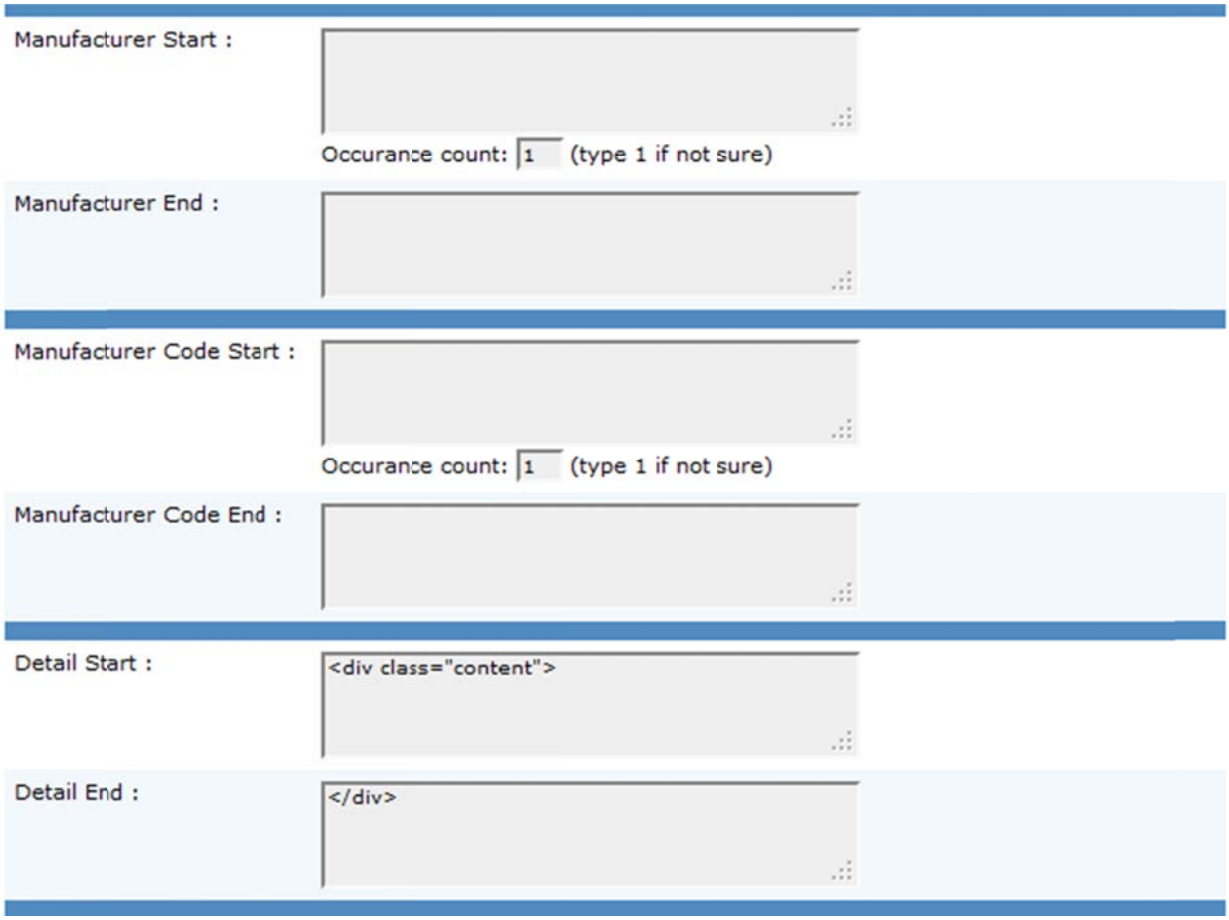
Category Start and Category End: This is similar to chunk start/end. The crawler needs this to identify the category from the product page.



Product Title Start and Product Title End: This is also similar to chunk start/end. It is used to extract the product title from the chunk. The only thing you see different is the "Occurance count". This is used to help find the correct product title from chunk if the "Product Title Start" tags are not unique. For eg, if you write <td> as start tag, this tag may occur many times in the chunk, and normally the crawler takes the first occurrence, but if that is not what you want, then you can specify that which occurrence of <td> tags begins the product title.

For e.g. in the following chunk, the third occurrence is the product title, so we should write 3 in the Occurance count text box:

<table class="nav_table">
<tbody>
<tr>
<td>Home : <a href="/cat/electronics__111"
class="breadcrumb">Electronics : <a href="/cat/cameras__113"
class="breadcrumb">Cameras</td>
<td>Price $551.99</td>
<td> JVC Everio HD Flash Memory Digital Camcorder</td>



Manufacturer Start and Manufacturer End: This will extract the brand of product just the way Product Title Start/End extracts the product name. If you leave any of these empty, then it will not extract the field value.

Manufacturer Code Start and Manufacturer Code End: This will extract the sku/part no of product just the way Product Title Start/End extracts the product name. If you leave any of these empty, then it will not extract the field value.

Detail Start and Detail End: This will extract the description of product just the way Product Title Start/End extracts the product name. If you leave any of these empty, then it will not extract the field value.

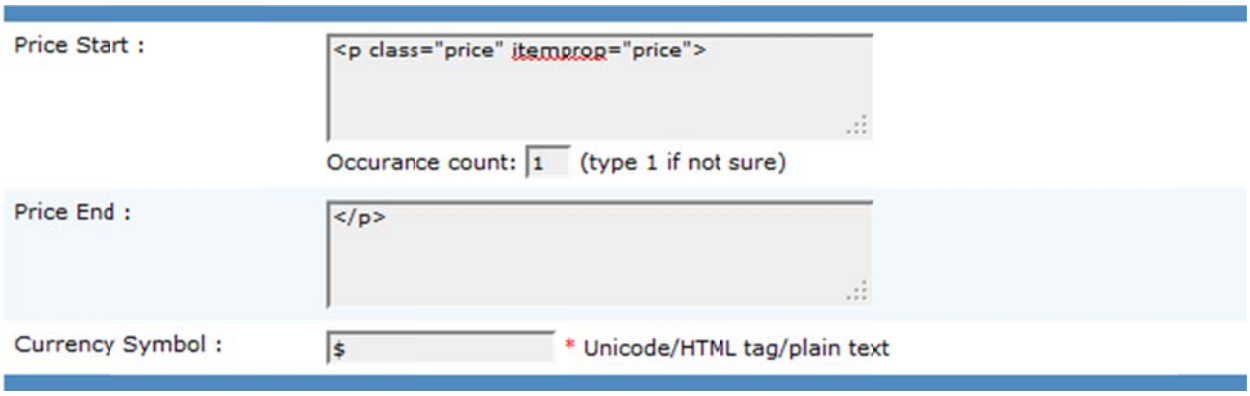
Image Url: This section helps crawler extract the product image from the remote server. Each image on server has some unique id that helps identify the product for which this image is. To tell that unique portion we use [IMAGEID] in the main Image Url field and below that we tell the crawler how to extract this [IMAGEID] from the product page chunk.

For e.g. a typical image url could be **http://image.bizrate.com/resize?sq=400&uid=[IMAGEID].jpg** where [IMAGEID] could be extracted from chunk by providing the wrapper content around that product's image url, which could be for e.g. **http://image.bizrate.com/resize?sq=100&uid=** and **.jpg** Note that the urls are different, the actual image url has sq=400 (for a larger size) while the url in the chunk has sq=100 to show a 100px width image on the product page. To find the actual large image url, you may need to click the product image on the product page of the merchant website and then see the url of the image that opens up (normally in a larger size).

In our case, the whole image url was changing for each product, because target.com uses different servers (like img2.targetimg2.com, img1.targetimg1.com, etc), so we placed [IMAGEID] for the whole changing portion of the image url and used the common portion as wrapper to find the value of [IMAGEID].

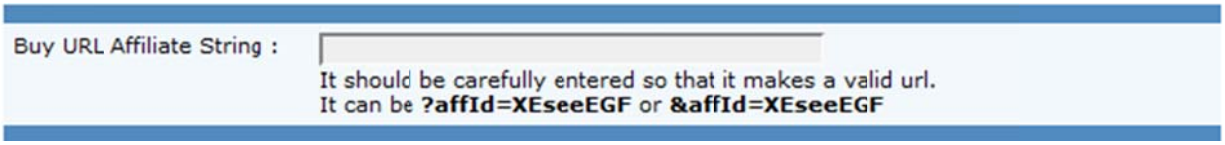
Comment Start and Comment End: This will extract the merchant comments just the way Product Title Start/End extracts the product name. If you leave any of these empty, then it will not extract the field value.

Shipping Start and Shipping End: This will extract the product shipping just the way Product Title Start/End extracts the product name. If you leave any of these empty, then it will not extract the field value.

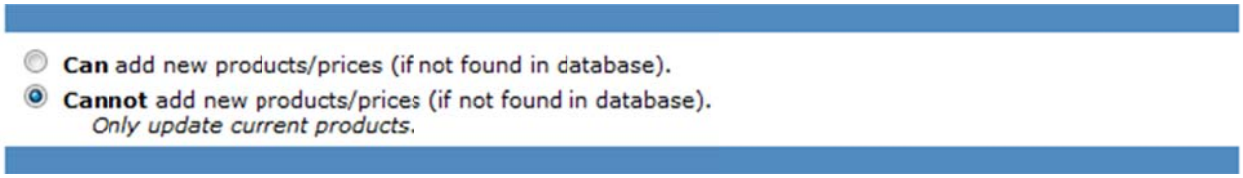


Price Start and Price End: This will extract the product price just the way Product Title Start/End extracts the product name. If you leave any of these empty, then it will not extract the field value.

Currency Symbol: You should provide the exact value of currency symbol used on merchant site in their html. This value is than excluded from the price we extract to get actual integral value of price.



Buy URL Affiliate String: This is a phrase that you might want to add at the end of the redirecting url achieved from the merchant site. This phrase can identify you as the affiliate sending traffic to the merchant site.

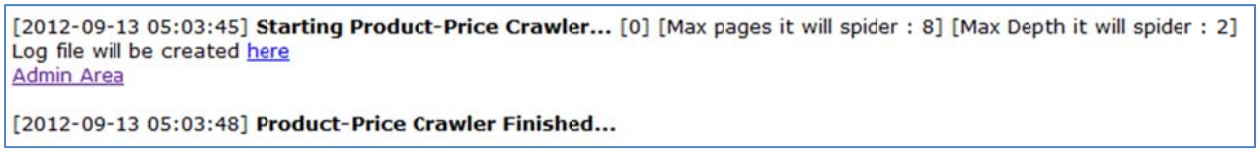


This is a new feature introduced in version 8 of the script. You can decide whether to add new products or only update old. This feature will be handy if you only want to update prices.

Start Product/Price Crawler:

After we have setup the crawler, we can execute it from Content > Categories > Manage Product/Price Crawler > Start Product /Price Crawler.

Once you hit that link, you will see the following processing page:



When the crawler finishes, click the log file link and you will see a page like this:

The crawler does not actually import the products extracted, but saves them as a csv file on your server for that merchant.

Now you can click the link highlighted in red below to start importing those products:

Or you can use the link highlighted in green to see the csv file created.