

How To

- [Debugging product prices](#)
- [Modifying stock levels \(for e.g. all stock -1\)](#)
- [Modifying attribute values and using Formulas](#)

Debugging product prices

There are a few positions which need to be checked. Read this article first to understand how product price updates work. In general:

1. We collect prices from your suppliers / product feeds
2. Based on "unique field" in the export profile (usually EAN code) we compare similar products and choose the cheapest product as base price
3. After we have determined product base price, pricing rules are applied and price markup / discount is added

Please note that repricing rules can be individual per export profile. Make sure you check which pricing rules are applied to your specific export profile.

So, let's say you suspect that exported product price is not the way it should be. Navigate to Tools → Product Price Checker and use "Search Criteria" form to enter product details. After you click "Search" - product information will appear on the right side.

Search Criteria

Use this form to check what pricing rules are applied for your products. This is useful when you want to make sure your pricing formulas and conditions work the way you want and that product prices that are exported to your sales channels are correct.

Supplier

ACC Distribution

Attribute

EAN

Attribute Value

5024442932205

Export Profile

ACC products (id: 205)

Search

Product Information

Supplier Name	ACC Distribution
Supplier Code	acc
Product Name	Targus Classic Fits up to size 16 ", Black, Backpack, Shoulder strap
Manufacturer	undefined
Product SKU	097234
EAN Code	5024442932205
Entity PK	097234
Quantity in Stock	0
Product Price	21.6377
Synced with Supplier ⓘ	2020-05-05 13:03:43

Applied Pricing Rules for Export Profile "ACC products (id: 205)"

Rule #1	
Rule Name	All +10%
Rule ID	272
Attribute	price
Formula	{attr:price}*1.05
Starting Price	21.6377
Calculation	21.6377*1.05 = 22.719585

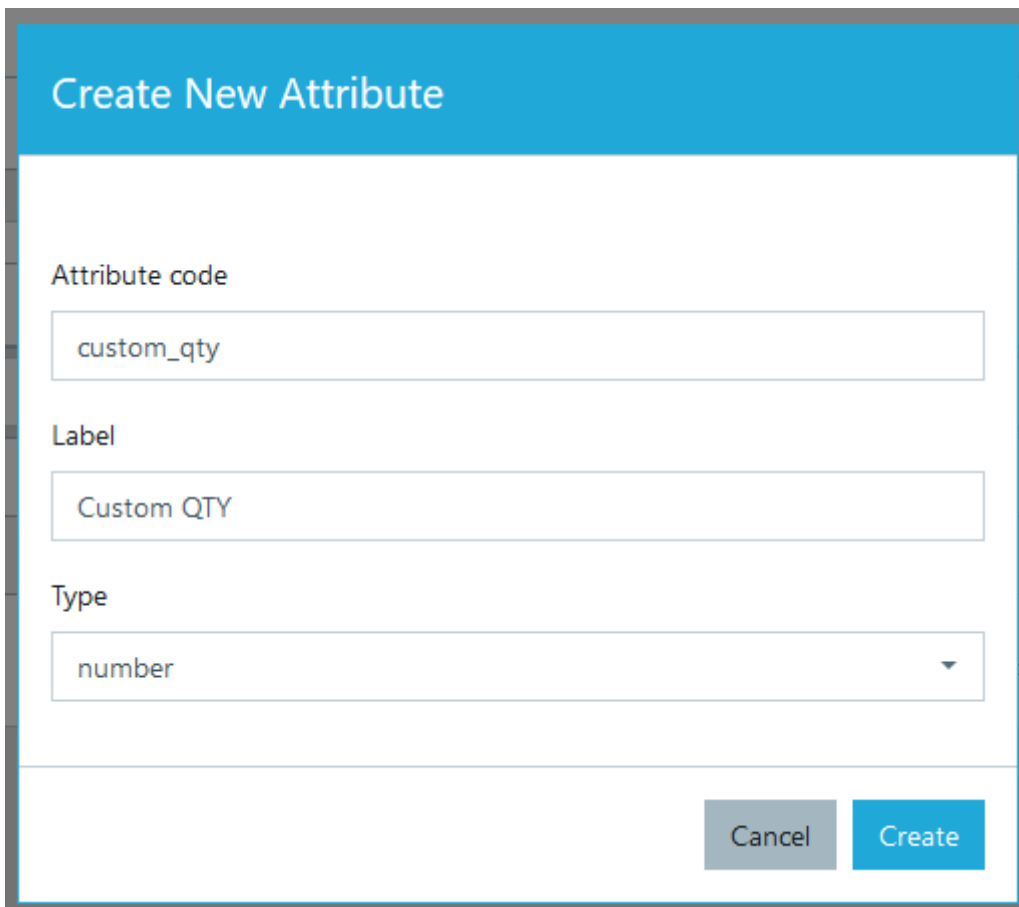
There can be multiple rules applied to the product. This example above shows that Pricing rule #205 is applied, product base / starting price was 21.6337 and the output result is 22.719585.

Modifying stock levels (for e.g. all stock -1)

In this article we will explain how to modify exported stock levels. This is useful when you want to reduce the risks of overselling. For example, if your supplier has 5 items in stock, but you want to export 4. Or if they have 1 item, then you export 0.

Step 1. Create a dynamic attribute

We will use the dynamic attribute which will hold the custom stock values. Navigate to "Dynamic attributes" and click "Add attribute". Lets give this attribute a code `custom_qty` and select type "number":

A screenshot of a web form titled "Create New Attribute". The form has a blue header bar with the title in white. Below the header, there are three input fields. The first is labeled "Attribute code" and contains the text "custom_qty". The second is labeled "Label" and contains the text "Custom QTY". The third is labeled "Type" and is a dropdown menu with "number" selected. At the bottom right of the form, there are two buttons: a grey "Cancel" button and a blue "Create" button.

Create New Attribute

Attribute code










Label

Type

Click create to save this selection.

Step 2. Assign dynamic attribute values

We can now assign values to this dynamic attribute. Click on a green icon "Values list":


ID	Label	Type	Actions
collectionhours	Collection Hours	number	  
zero_stock	Zero Stock	number	  
custom_qty	Custom QTY	number	  

Total page count: 1 Items per page: 300 Total items count: 3

New window will open. Click "Add value" and a pop-up will open. Make selection as in the example below (if product stock is greater or equal to 1, than `custom_qty` will be reduced by 1):

Create New Value

Conditions

AND	OR			+ Add rule	+ Add group
		stock	greater or equal	1	

Value

Priority

Cancel

Create

Click "Create" and your new value rule will be saved. Create one more value as a fallback - in case quantity from supplier is less or equal to 1, we will assign 0:

Create New Value

Conditions

AND OR

+ Add rule + Add group

Value





0

Priority

999

Cancel Create

Note the priority field - it has to be higher than the the first value we created. Click "Create" to save this value and your whole values list would look something like this:

Value	Priority	Actions
{attr:qty}-1	1	 
0	999	 

Total page count: 1 Items per page: 300 Total items count: 2

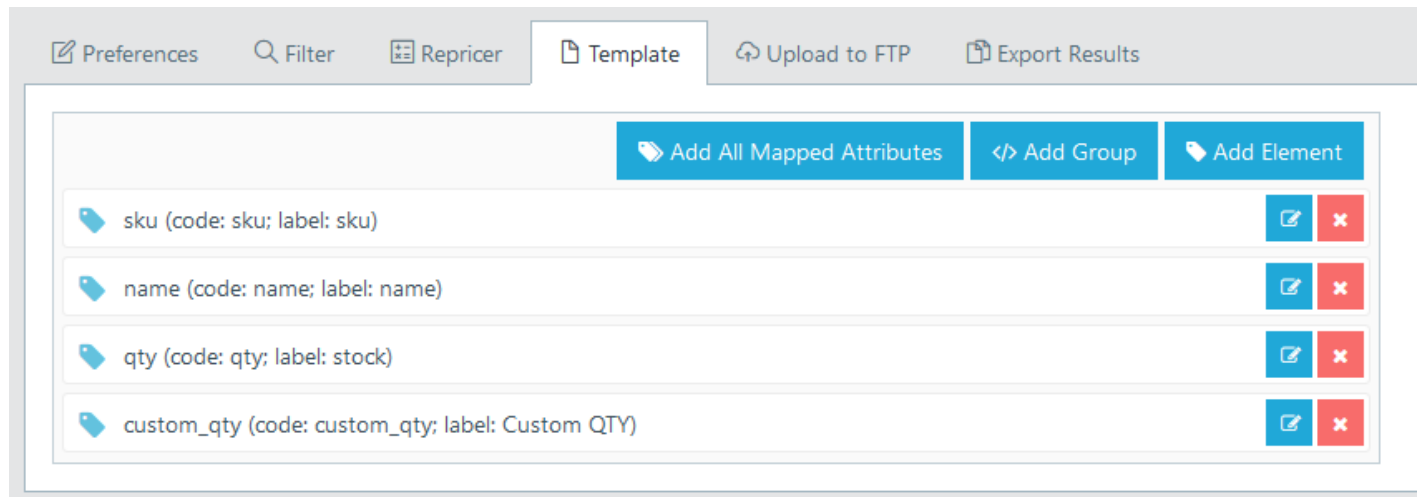
That is it - we now have the dynamic attribute with reduced stock levels by 1.

Step 3. Test it (optional)

Before you start exporting this newly created attribute, we recommend to test and make sure it works the way you wanted.

Navigate to "Export profiles" and create a new export profile with type XML / CSV / JSON. Select

your preferred output format and create a simple export template similar to this:



This will allow to see original and modified stock side by side. Save the template and execute it (add it to queue). Once it's status becomes "Completed" - open the generated file in the results tab.

See our example CSV. Columns C and D - `custom_qty` has smaller value than `qty`.

	A	B	C	D
1	sku	name	qty	custom_qty
2	PFC51B90CF5AF2AF25	Product name #1	6	5
3	PFF741199B1A8EB2179	Product name #2	6	5
4	PF311EF1DBEF8B015B	Product name #3	6	5
5	PF542D376A63A5B9571	Product name #4	6	5
6	PFDC1A5422C59820CE	Product name #5	6	5
7	PFF397743910DF215DE	Product name #6	6	5
8	PF3124F82EAE0D93D3	Product name #7	6	5
9	PFD5CEA987FB055CE6	Product name #8	6	5
10	PF6F770D2260B1AD6D2	Product name #9	5	4
11	PFB08A17623A049EB1	Product name #10	6	5
12	PFCF48DC1B013D171F	Product name #11	6	5
13	PF765A992305C1182A0	Product name #12	6	5
14	PF91FFE3F40B4204527	Product name #13	6	5
15	PF6555F26E1A985D5C7	Product name #14	6	5
16	PF730EA953355C0B1E0	Product name #15	6	5
17	PFFA9FD5F15972CC9B	Product name #16	6	5
18	PFAD08DD318B11EC55	Product name #17	6	5
19	PFF054006318BB281C3	Product name #18	6	5
20	PF3161030980159323A5	Product name #19	6	5
21	PFA2B7A8D3978E43263	Product name #20	6	5
22	PF0BD01E7C8A7F22879	Product name #21	6	5
23	PF1989F552FF28152018	Product name #22	6	5
24	PF46A217A2D5FAF463A	Product name #23	6	5
25	PF9880A35CED3E0A7A	Product name #24	6	5
26	PF4A199DAF0DE08789B	Product name #25	6	5
27	PF6C2BA5E22B7D7FEA	Product name #26	6	5
28	PF81883745765F8BD980	Product name #27	6	5

Step 4. Add "Custom QTY" to your export profile template

Open you preferred export profile and add this `custom_qty` attribute to the template (most often you will want to use this instead of the original `qty`).

That is it! If you have any questions - do not hesitate to contact us.

Modifying attribute values and using Formulas

As you have noticed, almost every product attribute can be accessed in [export templates](#) or [dynamic attributes](#) with an expression like `{attr:attribute_code}`. In this section we will explain, how you can modify attribute values or make some manipulation with them.

Retrieving the N-th element of an Array

Some attributes can have multiple values (for example one product can be assigned to multiple categories). Depending on your Export profile template, our app will automatically select first category or do some needed modifications.

However, if you use the static file export (CSV / XML / JSON), you can set to export only the N-th category. If attribute is not array (for example string, integer), it will skip modification.

Here are a few examples:

`{attr:categories|[0]}` - will return the first element of an array

`{attr:names|[english]}` - you can get array element by key

`{attr:names|[en]|[label]}` - multidimensional arrays are also supported

Serialization to JSON

`{attr:categories|json}` - if Categories value is array, then this modifier will convert it to JSON string

Value type conversion

Wise2Sync mainly holds two types of supplier attributes: numeric and text. You can switch between types like this:

`{attr:qty|txt}` - Typecast conversion to String `|txt`. Value will be converted to String type

`{attr:qty|int}`

- Typecast conversion to Int `|int` . Value will be converted to Int type
- `{attr:length|float}` - Typecast conversion to Float `|float[:precision]` . For example `{attr:length|float:2}` will convert Length value to Float type with precision of 2.

Examples

Create New Value

Conditions

AND OR

+ Add rule

+ Add group

Value

{attr:length|float}/100 * {attr:height|float}/100 * {attr:width|float}/100

Priority

1

Cancel

Create

#1 Product volume. Lets say, you want to calculate total product volume in cubic meters, when you supplier only provides with, height, and length in centimeters.

What you do is create a dynamic attribute with type "number" and set a value expression:

```
{attr:length|float}/100 * {attr:height|float}/100 * {attr:width|float}/100
```

#2 Custom product name. Lets say you would like to have have custom product names, which would include product Category label as a prefix. So, what you would do is create a textual dynamic attribute "Custom Name" and set its value as follows:

```
{attr:lab_val_category|[0]} - {attr:name}
```

So if your product name was "Funny Sunshine 101" and it belongs to a category "Eye Glasses", then product "Custom name" would be "Eye Glasses - Funny Sunshine 101".

Support across the Wise2Sync App

Type	Expression	Example	Dynamic Attributes	Export profile Templates	Repricing Rules
Array element value	<code> [n]</code>	<code>{attr:category [0]}</code>	Yes	Yes	No

Array serialization to JSON	<code> json</code>	<code>{attr:tags json}</code>	Yes	Yes	No
Value type conversion	<code> txt</code> , <code> int</code> , <code> float[:precision]</code>	<code>{attr:length float:2}</code>	Yes	Yes	No
Numeric operators	<code>ceil(..)</code> , <code>floor(..)</code> , <code>round(..)</code> , <code>abs(..)</code>	<code>ceil({attr:length float})</code>	Yes	Yes	Yes
Find lowest / highest value	<code>min(..)</code> , <code>max(..)</code>	<code>min({attr:price}, 0)</code>	Yes	Yes	Yes
Current Date	<code>{date:format}</code>	<code>{date:Y-m-d}</code>	Yes	Yes	No