

foxy feeds

A stylized orange pencil graphic is positioned below the word "foxy". The pencil is oriented horizontally, pointing to the right. The body of the pencil is a solid orange color, and the tip is a lighter shade of orange. The pencil is positioned such that its body extends under the letters 'f', 'o', and 'x' of "foxy", and its tip points towards the letter 'y'.

Foxy Feeds uses formulas that are based on syntaxes that are similar to those used in programming. These formulas can be used to modify imported data fields and to create custom export filters.

### YOUR DATA FIELDS

Use preexisting data fields from your product catalogue to create custom data fields.

Filter

CREATE NEW GROUP

1/2 1/2

▲ No group assigned (6)

- Category Custom
- Clean HTML
- Colors
- Price
- Stock availability
- Title Custom
- Apply own data field

### YOUR PRODUCT FILTER

Set up your product filter to control which products will be exported to which channel. The filter set up can be used for the shop settings in general or for channel-specific settings for certain channels.

Filter

CREATE NEW GROUP

1/2 1/2

▲ No group assigned (2)

- Inventory < 5
- North Ridge
- Add filter

Custom data fields enable you to create additional fields required by a network that is not available in your current import catalogue.

These formulas are used to generate or process data from your current product catalogue.

Export filters are used to segment your exported products according to predefined criteria. The filters can be applied as either a whitelist or blacklist with products being specifically pushed to a channel or held from being exported to a channel respectively.

## Creating a filter or custom data field

STOCK AVAILABILITY

CLOSE X

Settings

X DELETE

NAME ?

TYPE OF DATA ?

DATA FIELD ?

PREVIEW

Article11B-2500002488854

quant 2

Number 3

Quantity

5

Select data field

1

FORMULA ?

if(quant&gt;0,"Available","Not available") 4

RESULT PREVIEW

Available 6

TEST IN FULL

5

TEST

Define fixed article numb

SAVE SETTINGS

1. Select the data field(s) to be used in the formula
2. Complete the name of each parameter to be used
3. Select the type of data included in the field (text or number)
4. Enter your formula
5. Select test (single entry) or full test (whole catalogue)
6. See overview of test result

## Example

Formula/Function	Parameters			Remarks
	Parameters	Formula	Result	
<p>a + b + c</p> <p>Combining datafields without spaces</p>	<p>Cat Brand Gender</p>	<p>Cat + Brand + Gender</p>	<p>BasketsNikeFemale</p>	
<p>a + " " + b + " " + c</p> <p>Combining datafields with spaces</p>	<p>Cat Brand Gender</p>	<p>Cat + " " + Brand + " " + Gender</p>	<p>Baskets Nike Female</p>	<p>Includes a space between datafields</p>
<p>a + " - " + b + " - " + c</p> <p>Addition of multiple fields of data or numbers, separated by spaces and hyphens</p>	<p>Cat Brand Gender</p>	<p>Cat+ " - " + Brand + " - " + Gender</p>	<p>Basket - Nike - Femme</p>	<p>Includes a „ - „ between datafields</p>

## Example

Formula/Function	Parameters	Formula	Result
<p>Available in "" + a + "."</p> <p>Addition of several fields of the catalogue data + additional text of punctuation</p>	Size	"Available in" + Size + "."	Available in 36-41.
<p>a + "-" + b + " for" + c + "/" + d + "-" - Size : "+ e + "." + f</p> <p>Addition of several data fields of the catalogue + added text of punctutation.</p>	Category Brand Gender Colour Size Description	categorie + "-" + brand + » for " + gender + "/" + colour + "-" - Size: » + "." + + description	Baskets – Nike for Women // Black – Size 38-39-40. Trendy urban womens shoes

## Example

Formula/Function	Parameters	Formula	Result
avg(a, b, c, ...) Calculate an average	a:597 b:130 c:2	avg(597, 130, 2)	243
min(a, b, c, ...) Calculate a minimum	a:597 b:130 c:2 d:469	min(597, 130, 2, 469)	2
max(a, b, c, ...) Search for a maximum value	a:597 b:130 c:2 d:469	max(597, 130, 2, 469)	597

## Example

Function/Formula	Example			Remarks
	Parameters	Formula	Results	
ceil(a) Round up to the next number	a: 2,1465	ceil(2.1465)	3	A single parameter may be used, the decimal should be marked using a point and not a comma
floor(a) Round down to the next number	a:2,7	floor(2.7)	2	A single parameter may be used, the decimal should be marked using a point and not a comma
round(a,b) Round the parameter „a“ according to the number of decimal places „b“	a: 5.3647 b: 2	round(5.3647,2)	5.36	



## Example

Formula/Function	Parameter			Remarks
	Parameter	Formula	Result	
if(cond, valtrue, valfalse)	Conditions : - price >= to 50 - valtrue: 0.00, - valfalse: 5.95	if(price >= 50, "0.00", "5.95")	0.00 or 5.95	This formula can be used for example to apply free delivery is a product costs a certain amount.
The „If“ function is a logical function. It allows a return value if a condition is true or false.	Conditions : -Brand: Nike, valtrue:desc(descripti on), valfalse: "" („Nothing“).	If(brand == "Nike", "desc", "")	24H only : -20% discount on Nike shoes Nike Code: Nike20	If the description is displayed then the condition is „true“. If the condition is „false“ then it will not appear.
case(a, b, c, d, e, ..., y)  Verifies and applies an action if the value (b,c,d,e....) in the parameter is checked	Conditions: A:colour b:kaki c:vert d:mauve e:violet Y:colour	case(colour, "kaki", "vert", "mauve", "violet", colour)	"Vert" or "Violet"	In this example, the formula for changing existing colours. If the colour (a) is "kaki", replace it with "vert" (b). If the colour (c) is "mauve" replace with "violet" (d). If it is not either of these colours keep the existing colour.

**Example**

Formula/Function	Example			Remarks
	Parameters	Formula	Results	
<p>to_text(a)</p> <p>Converts numbers to text</p>	<p>a: 2.95</p>	<p>to_text(2.95) + "€"</p>	<p>2.95€</p>	
<p>left(a,b)</p> <p>Select a number of characters or digits from the left</p>	<p>a: desc b:177</p>	<p>left(desc, 177)</p>	<p>Beautiful floral print skirt in black and turquoise</p>	<p>Only the first 72 characters from the left will be displayed. Attention: punctuation marks are also considered characters.</p>
<p>right(a,b)</p> <p>Select a number of characters or digits from the left</p>	<p>a: title b:15</p>	<p>right(title, 15)</p>	<p>C3590 Red Original</p>	<p>The base title is "Samsung C3590 Red Original"</p>
<p>substr(a, b, c)</p> <p>Select only a part of a text "a" between "b" and "c".</p>	<p>a:"Telephone" b:2 c: 4</p>	<p>substr("Telephone", 2, 4)</p>	<p>le</p>	

Formula/Function	Example		
	Parameters	Formula	Result
upper(a) Converts characters to upper case	a: fabric	upper(fab)	NIKE
lower(a) Converts characters to lower case	a: brand	lower("Nike")	nike
len(a) Counts the number of characters in the text "a"	a:Chaussure	len("Chaussures")	10
capitalize(a, b) Converts the first letter of each word to upper case	a: New Nike Air Max Shoes b: "true" or "false"	capitalize("New Nike Air Max Shoes", true)	New Nike Air Max Shoes
url_encode(a) Converts text to URL code text	a: lien	url_encode(lien)	http%3A%2F%2Fexemple.fr%2Fa%2F%3F%3Dcpc%26client%3D%26l%3Dfr%26camp%3Ddeep%26nw%3Dsem%26deep

Formula/Function	Example			Remarks
	Parameters	Formula	Results	
trim(a) Removes Spaces before and after text	desc(description)	trim(" desc ") trim(" ma description ")	" ma description "	
search_and_cut(a, b) Search for a value and then cut the text from the first instance	Ex: a:"New and Improved!!" b:"et"	search_and_cut("New and Improved", "and")	New	In this example, the search value is „and“. From the first instance of the word the text is cut.
clean_html(a) Removes HTML tags from text	Ex: "<head>my text</head>"	clean_html(<head>my text</head>)	My Text	
replace(a, b, c) Searches for a value and replaces it	a:title b:colour c:"Customisable"	replace(title, colour, "Available in blue or green")	Nike Air Max// Blazer Collector 1980 red, available in blue or green	Search in column „a“, the value „b“ and replace it with „c“. In this example the colour is replaced by the text: „Available in blue or green“.
in_string(a, b) Checks as to whether an expression exists within a data field	a:"Black t-shirt mens" b:"Men Women"	in_string("Black t-shirt mens", "Men  Women")	"false"	Value of „false“ indicates that the expression „a“ does not exist in the data field „b“.
in_list(a, b, c) Checks for value „a“ in list „b“ with seperators „c“	a: "123" b: "123,190" c: ","	in_list("123", "123,190", ",")	"true"	

## Example

Formula/Function	Example			Remarks
	Parameters	Formula	Result	
<b>is_number(a)</b> Checks if value is a number	a: price	is_number(prix)	"true"	Price is expressed as a number hence the result „true“ where description is expressed as a text hence the result „false“
	a: description	is_number(description)	"false"	
<b>to_number(a, b)</b> Converts text to a number	a: "A8" b: true	to_number("A8", true)	8	
<b>base64_encode(a, b)</b> Encodes text in base64	a: description ex"Nike sports shoe" b: est "UTF-8"	base64_encode("description", "UTF-8")	TGUgY2hhdCBtaWF1bGUslG1haXMgbGUgY2hpZW4gYWJvaWU=	
	a: "TGUgamFyZGluaWVylGphcmRpbmUsIG1haXMgbGUgYm91bGFuZ2VylIG5lIGJvdWxhbmdlIHhwcw==" b:UTF-8	base64_decode("TGUgamFyZGluaWVylGphcmRpbmUsIG1haXMgbGUgYm91bGFuZ2VylIG5lIGJvdWxhbmdlIHhwcw==", "UTF-8")	My text is available here	
<b>replace_regexp(a, regexp, c)</b> Searches in regular expression data fields „a“ and replaces with „c“	a: Colour Regexp: "Noir Mauve Kaki Rose" c: "."	replace_regexp(colour, "Noir Mauve Kaki Rose", ".")	.	In this example, the formula is used to override the values.

## Example

Formula/Function	Example			Remarks
	Parameters	Formula	Result	
<p><code>search_and_cut_back_v2(a, b)</code></p> <p>Deletes part of text „a“ from point „b“</p>	<p>a: "New and Improved!!" b: "and"</p>	<p><code>search_and_cut_back_v2("New and Improved", "and")</code></p>	<p>Improved</p>	<p>In this example, the search value is "and". All that is before the text will be removed.</p>
<p><code>in_string_regexp(a, regexp)</code></p> <p>Checks whether one or more expressions exist within another</p>	<p>a: "Men's t-shirt black" regexp: "Mens   Women"</p>	<p><code>in_string_regexp("Men's t-shirt black", "Mens   Women")</code></p>	<p>"true"</p>	<p>The value of "true" indicates that at least one occurrence has been found in the data field.</p>
<p><code>category(a, b, c, d ...)</code></p> <p>Creates category tree with separator</p>	<p>a: "&gt;" b: "Mens" c: "Brands" d: "Nike"</p>	<p><code>category("&gt;", "Men's", "Brands", "Nike")</code></p>	<p>Mens &gt; Brands &gt; Nike</p>	