

ARRAYSTOMAP Function

Contents:

- *Basic Usage*
- *Syntax and Arguments*
 - *array_keys*
 - *array_values*
 - *DEFAULT KEY*
- *Examples*
 - *Example - Create an Object of product properties*

Combines one array containing keys and another array containing values into an Object of key-value pairs.

- This function applies to two inputs only.
- Inputs can be array literals, column references, or functions returning arrays.

If the number of key elements is greater than the number of value elements, null values are generated for the missing values in the output Object. If the number of value elements is greater, the `DEFAULT_KEY` value (third parameter) is applied.

Basic Usage

Array literal reference example:

```
derive type:single value:ARRAYSTOMAP(["A","B"],["1","2","3"]) as:'myMap'
```

Output: Generates an Object associating keys from the first array with values from the second array.

Column reference example:

```
derive type:single value:ARRAYSTOMAP(array1,array2, 'extraProps') as:'array2mapped'
```

Output: Generates a new `array2mapped` column containing an Object pairing the elements of the arrays as key-value pairs. Any extra values in `array2` are assigned to the `extraProps` key.

Function reference example:

```
derive type:single value:ARRAYSTOMAP(array1,concat([array2,array3])) as:'array2map'
```

Output: Generates a new `array2map` column containing an Object pairing the elements of `array1` and the array created by concatenating `array2` and `array3`.

Syntax and Arguments

```
derive type:single value:ARRAYSTOMAP(array_keys,array_values, ['DEFAULT KEY'])
```

Argument	Required?	Data Type	Description
array_keys	Y	string or array	Name of column, array literal, or function returning an array whose elements are the keys for the generated Object

array_values	Y	string or array	Name of column, array literal, or function returning an array whose elements are the values for the generated Object
DEFAULT_KEY	N	string literal	Any extra values are assigned to this specified key

For more information on syntax standards, see *Language Documentation Syntax Notes*.

array_keys

Name of the array column, array literal, or function returning an array whose elements you want to use as the keys for the Object.

Usage Notes:

Required?	Data Type	Example Value
Yes	String (column reference, function returning an array) or array literal	myKeys

array_values

Name of the array column, array literal, or function returning an array whose elements you want to use as the values in the Object.

Usage Notes:

Required?	Data Type	Example Value
Yes	String (column reference, function returning an array) or array literal	myValues

DEFAULT KEY

If there are extra elements in the second array, they are assigned to the key that is defined by this parameter.

Usage Notes:

Required?	Data Type	Example Value
No	String literal	'extraProperties'

Examples

 **Tip:** For additional examples, see *Common Tasks*.

Example - Create an Object of product properties

Source:

Your dataset contains master product data with product properties stored in two arrays of keys and values.

ProdId	ProdCategory	ProdName	ProdKeys	ProdProperties
S001	Shirts	Crew Neck T-Shirt	["type", "color", "fabric", "sizes"]	["crew", "blue", "cotton", "S,M,L", "in stock", "padded"]
S002	Shirts	V-Neck T-Shirt	["type", "color", "fabric", "sizes"]	["v-neck", "white", "blend", "S,M,L,XL", "in stock", "discount - seasonal"]

S003	Shirts	Tanktop	["type", "color", "fabric", "sizes"]	["tank","red","mesh","XS,S,M","discount - clearance","in stock"]
S004	Shirts	Turtleneck	["type", "color", "fabric", "sizes"]	["turtle","black","cotton","M,L,XL","out of stock","padded"]

Transform:

When the above data is loaded into the Transformer page, you might need to clean up the two array columns.

Using the following transform, you can map the first element of the first array as a key for the first element of the second, which is its value. You might notice that the number of keys and the number of values are not consistent. For the extra elements in the second array, the default key of `ProdMiscProperties` is used:

```
derive type:single value: ARRAYSTOMAP(ProdProperties, ProdValues, 'ProdMiscProperties')
as: 'prodPropertyMap'
```

You can now use the following steps to generate a new version of the keys:

```
drop col:ProdKeys
```

```
derive type:single value:KEYS(prodPropertyMap) as:'ProdKeys'
```

Results:

ProdId	ProdCategory	ProdName	ProdKeys	ProdProperties	prodPropertyMap
S001	Shirts	Crew Neck T-Shirt	["type", "color", "fabric", "sizes", "ProdMiscProperties"]	["crew","blue","cotton","S,M,L","in stock","padded"]	{ "type": ["crew"], "color": ["blue"], "fabric": ["cotton"], "sizes": ["S", "M", "L"], "ProdMiscProperties": ["in stock", "padded"] }
S002	Shirts	V-Neck T-Shirt	["type", "color", "fabric", "sizes", "ProdMiscProperties"]	["v-neck","white","blend","S,M,L,XL","in stock","discount - seasonal"]	{ "type": ["v-neck"], "color": ["white"], "fabric": ["blend"], "sizes": ["S", "M", "L", "XL"], "ProdMiscProperties": ["in stock", "discount - seasonal"] }

S003	Shirts	Tanktop	["type", "color", "fabric", "sizes", "ProdMiscProperties"]	["tank", "red", "mesh", "XS,S,M", "discount - clearance", "in stock"]	{ "type": ["tank"], "color": ["red"], "fabric": ["mesh"], "sizes": ["XS,S,M"], "ProdMiscProperties": ["discount - clearance", "in stock"] }
S004	Shirts	Turtleneck	["type", "color", "fabric", "sizes", "ProdMiscProperties"]	["turtle", "black", "cotton", "M,L,XL", "out of stock", "padded"]	{ "type": ["turtle"], "color": ["black"], "fabric": ["cotton"], "sizes": ["M,L,XL"], "ProdMiscProperties": ["out of stock", "padded"] }