Array Mapping

Regarding array mapping, all simple mapping features also apply - the only difference is that they can be applied to dedicated array elements.

Mapping the Complete Array

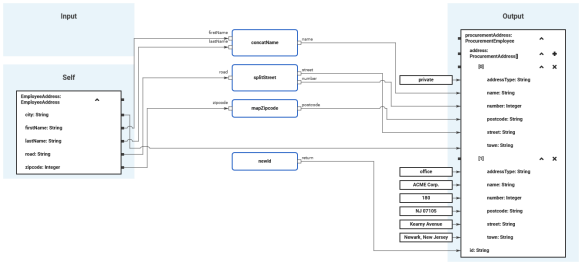
If the structures of the arrays are identical, you can directly connect source and target arrays with a relation. In this case, all array elements of the source array will be mapped to the target array.

Array_Data_Mapping_Example

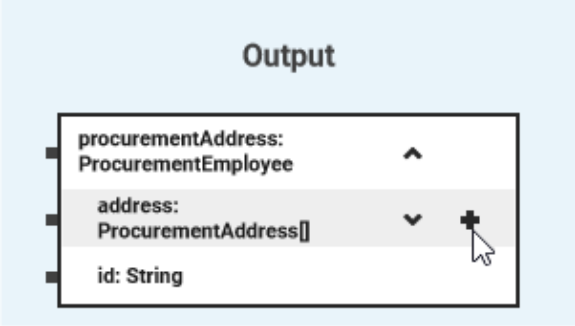
Click the icon to download a simple example model that shows how to implement array mappings in **Scheer PAS** *Designer*.

Mapping Dedicated Array Elements

If the structures of the source and target arrays differ, you can expand the arrays and map dedicated array elements. The above linked array example **Array_Data_Mapping_Example** contains a mapping diagram that shows a mapping of a (non-array) address to an address array.



You can map to dedicated array elements with a few simple steps:



Add an item to an array in a mapping diagram by clicking the ✚ icon to the right of the array's name.

On this Page:

Array_Data_Mapping_Example

Click the icon to download a simple example model that shows how to implement array mappings in **Scheer PAS** *Designer*.

Array_Mapping_With_Condition_Example

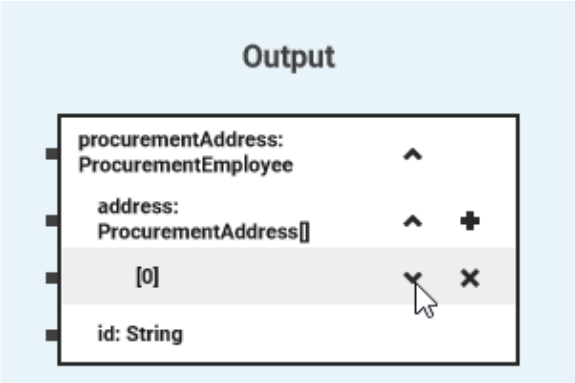
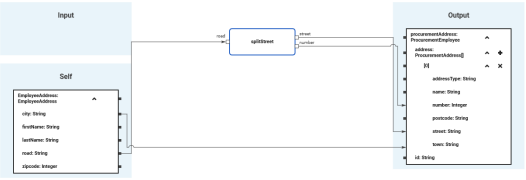
Click the icon to download a simple example model that shows how to extract data out of an input array using a condition as an index to access a dedicated array element in **Scheer PAS** *Designer*.

Array_Mapping_With_foreach_Example

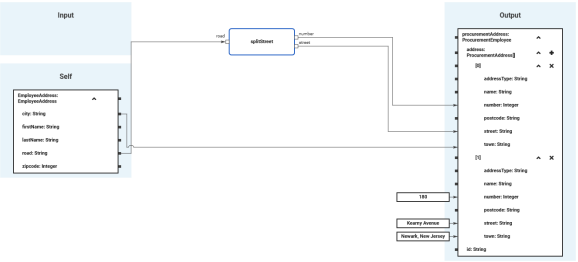
Click the icon to download a simple example model that shows how to map array content for each array element to a target array in **Scheer PAS** *Designer*.

Related Pages:

- Action Script Language

	A new array element is displayed. In this case, this is the first array element **[0]** as array numbering starts with 0. Expand the structure of the array element by clicking the little arrow to the right of the element number.
	Now, you can start drawing the mappings as usual.

You can add multiple array elements to the mapping structure if needed, to apply different mappings for each of them.



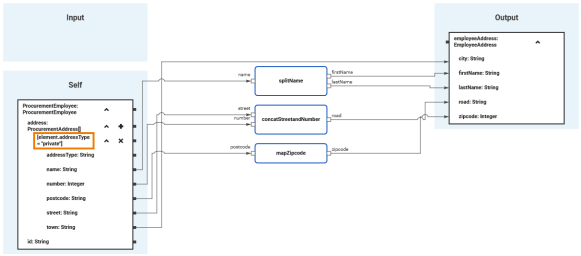
The picture above shows different mappings for street and town for the two array elements.

Conditional Mapping of Array Elements

Array_Mapping_With_Condition_Example

	Click the icon to download a simple example model that shows how to extract data out of an input array using a condition as an index to access a dedicated array element in **Scheer PAS** *Designer*.

If an input array contains multiple elements but you only want to process **one** element that can be identified by a dedicated attribute value, you can select this element using a condition in the index of the array.



In the example above, the first of all array element is selected that contains **private** as the content of attribute **addressType**. All other elements are omitted.

Expand the array, and simply double-click the index to enter a condition. Use keyword **element** to refer to an array element as opposed to a normal, absolute index , and build an action script expression that evaluates to **Boolean**.

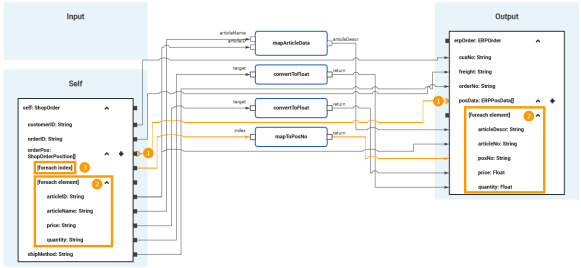
> The mapping condition finds the **first** element that matches the conditions. See also Limitation: foreach Mapping With Conditions.

Mapping Array Elements in a Loop (foreach)

Array_Mapping_With_foreach_Example

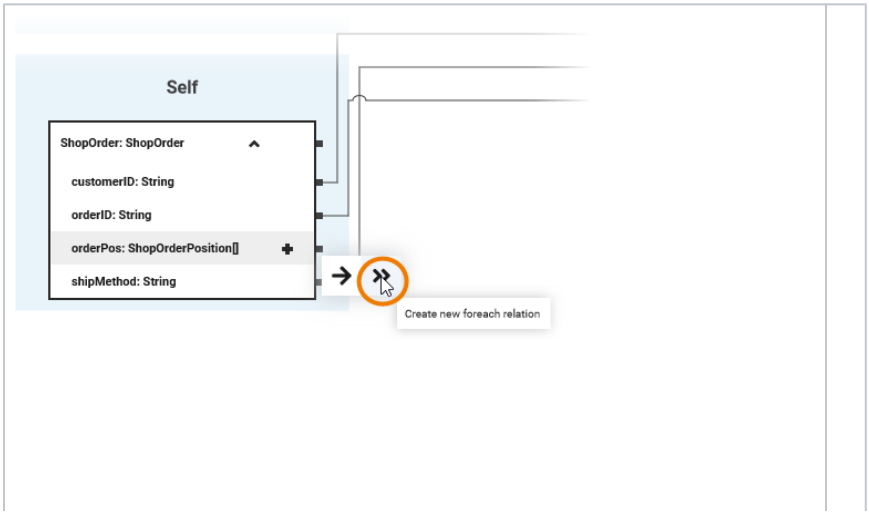
Click the icon to download a simple example model that shows how to map array content for each array element to a target array in **Scheer PAS** *Designer*.

You can define mappings that will be performed for each array element of the source array.



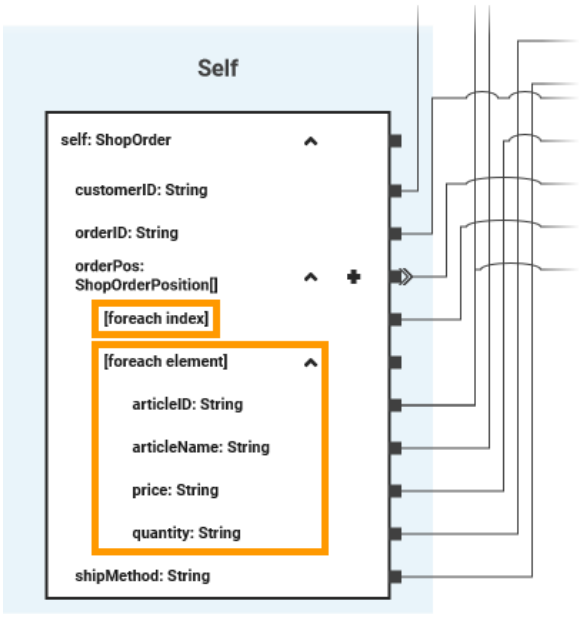
A for-each mapping consists of two steps:

1. **Link the arrays to be mapped**



You need to link the arrays in the input and output sections you want to map with a **foreach** relation (1).

	Click on the pin of the array to open the relation menu.



A **foreach index** (3) is created for the mapping source, and a **foreach element** (2) for each of the arrays.

2. **Map the array element properties**
   You now can map the array as described on Supported Mapping Functions. The defined mappings will be performed for every existing array element during process execution.
   - You can map a complete array element by connecting the **foreach elements** directly, or you can map dedicated array element properties as shown in the example above.
   - You can use **foreach index** to access the index of the currently mapped array element. In the mentioned example, the **foreach index** is mapped to a order position number in the target structure.

Please also note the limitation explained below.

Limitation: foreach Mapping With Conditions

The usage of foreach together with conditional mapping has limitations. The conditional mapping is limited to only return the **first** element that matches the condition.

So, keep the following in mind when using the two together:

- The foreach processes all elements of the source array and maps them to the target array.
- The conditions evaluates to the same array element (the first one that matches the condition) each time.

Result: The target array has the same count of elements as the source array. All elements contain the identical data of the one element that has been selected by the condition.