




# mapEqualNamesIfExists

<b>Syntax</b>	<pre>set aTargetObject = anInputObject.mapEqualNamesIfExists ([anotherInputObject] +); append mapEqualNamesIfExists(anInputObject) to anArrayOfTargetObjects;</pre>	
<b>Semantics</b>	<p>The macro works like <a href="#">mapEqualNames() Macro</a> if the mapped source attribute exists. If it doesn't exist, the mapping is skipped for this attribute. This way existing target attributes are not overwritten by NULL values of source attributes.</p> <div> You can supply multiple input objects.</div> <div> Only the existence of the <b>attribute values</b> is checked. It is assumed that the input <b>object</b> exists.</div> <div> This macro does not work recursively and thus do not perform a deep copy of attributes of complex types.</div>	
<b>Substitutables</b>	anInputObject, anotherInputObject	Can be any <b>complex</b> object having attributes of any type.
<b>Examples</b>	<pre>set person = person1.mapEqualNames(person2, person3); set person.address = mapEqualNames(address); set person.alternativeAddress = detailedPerson.address. mapEqualNames(); set person.address = mapEqualNames(detailedPerson.addresses [0]); append mapEqualNames(detailedPerson) to people;</pre>	

## On this Page:

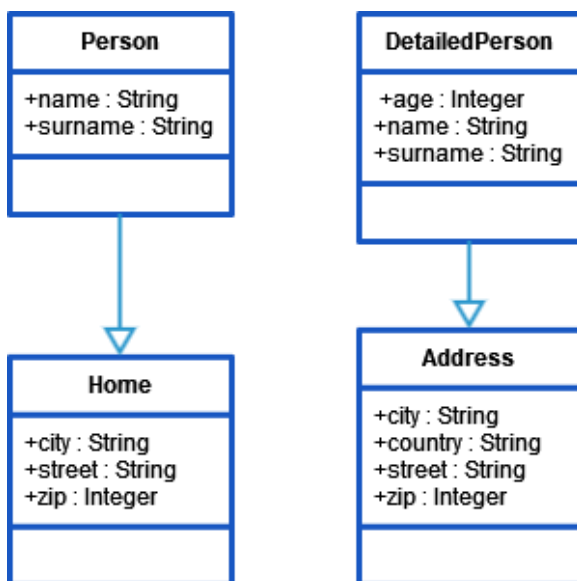
- [Usage of mapEqualNames\(\)](#)
- [Usage of "mapEqualNamesIfExists\(\)"](#)
- [Usage of mapEqualNames\(\) with the "append" Statement](#)

## Related Pages:

- [mapEqualNames\(\) Macro](#)

## Usage of mapEqualNames()

Given are the two unrelated classes [Person](#) and [DetailedPerson](#) .



Data / Script	Description / Result
<pre>personIn {   age: 45,   name: "Rose",   surname: "Bloom",   city: "San Francisco",   country: "USA",   street: "7, Waterfall Av.",   zip: 94016 }</pre>	<p>The object <b>personIn</b> is of type <b>DetailedPerson</b> .</p>
<pre>set personOut = mapEqualNames(personIn);</pre>	<p>This statement assigns values to the matching attributes of object <b>personOut</b> which is of type <b>Person</b> :</p> <pre>personOut {   name: "Rose",   surname: "Bloom",   city: "San Francisco",   street: "7, Waterfall Av.",   zip: 94016 }</pre> <p><b>age</b> and <b>country</b> remain unset.</p>

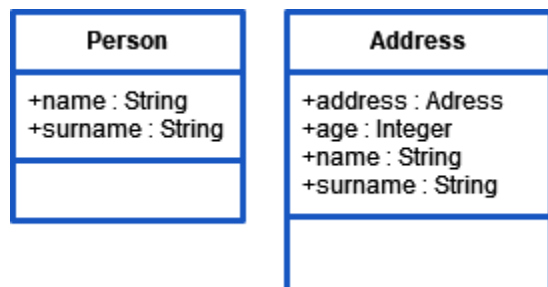
## Usage of "mapEqualNamesIfExists()"

Data / Script	Description / Result
<pre>personIn1 {   name: "John Robert Edward",   surname: "Snow",   city: "Anchorage",   street: "99, Malamute Street",   zip: 0 }</pre>	<p>The object <b>personIn1</b> of type <b>Person</b> already contains values. In our example, the zip code is unknown, so the attribute <b>zip</b> contains the value <b>0</b> .</p>
<pre>personIn2 {   age: 32,   name: "John",   surname: "Snow",   city: "Anchorage",   country: "USA",   zip: 12345 }</pre>	<p>The object <b>personIn2</b> is of type <b>DetailedPerson</b> . In this example object, the attribute <b>street</b> is not given.</p>
<pre>set personIn1 = mapEqualNamesIfExi sts(personIn2);</pre>	<p>This statement assigns values from <b>personIn2</b> to the matching attributes of <b>personIn1</b> :</p> <pre>personIn1 {   name: "John",   surname: "Snow",   city: "Anchorage",   <b>street: "99, Malamute Street",</b>   zip: 12345 }</pre> <p>The properties of <b>personIn1</b> get overwritten with existing values. The value of <b>street</b> remains unchanged, because it did not exist in the source object <b>personIn2</b> (was NULL).</p>

## Usage of mapEqualNames() with the "append" Statement

You can use `mapEqualNames()` along with the [append statement](#) to add a complex object with numerous attributes to an array of unrelated objects which needs only some of the information. The `mapEqualNames()` macro will create set statements for all equal named attributes found in the target object while the `append` statement will add the result to an array.

Given are the two unrelated classes **Person** and **DetailedPerson**.



Data / Script	Description / Result
<pre>aDetailedPerson {   adress: {     city: "San Francisco",     country: "USA",     street: "7, Waterfall Av.",     zip: 94016   }   age: 45,   name: "Rose",   surname: "Bloom" }</pre>	<p>Assume that <b>aDetailedPerson</b> is provided from an external source. For further processing, we are interested in <b>name</b> and <b>sur</b> <b>name</b> only.</p>
<pre>append mapEqualNames (aDetailedPerson) to customers;</pre>	<p><code>mapEqualNames()</code> collects the data from <b>aDetailedPerson</b>, and <code>append</code> adds it to the array <b>customers</b> . <b>customers</b> is an array of type <b>Person</b> and contains already the data of " <b>John Snow</b> " and " <b>Liv Falls</b> " .</p> <p>Result array:</p> <pre>customers [   {name: "John", surname: "Snow";}   {name: "Liv", surname: "Falls";}   {name: "Rose", surname: "Bloom";} ]</pre>