

# Creating Calculations

Calculated metrics allow you to create calculations based on the attributes of process steps or custom attributes. Various operators and expressions are available to create a calculation rule for a metric.

- You can calculate the duration between two steps.
- You can calculate with custom attributes (addition, subtraction).

 Currently it is only possible to offset **numbers** against each other.

 If you need further help with the implementation of your calculations, do not hesitate to contact your **Scheer PAS** consultant.

## Available Operators

The following table of operators can be searched. To do this, enter your search phrase in the search field:

Operator	Description	Structure
<b>OperatorAddition</b>	<ul style="list-style-type: none"><li>• Allows the addition of <b>expression2</b> and <b>expression1</b>.</li></ul>	<pre>"type": "OperatorAddition", "expression1": {}, "expression2": {}</pre>
<b>OperatorAttribute</b>	<ul style="list-style-type: none"><li>• Allows to extract an attribute (second operand) from the first node of a set of process nodes. This set of process nodes is itself the result of an expression (first operand).</li><li>• Possible values for <b>expression2</b>: start, end, name (of a process step).</li></ul>	<pre>"type": "OperatorAttribute", "expression1": {}, "expression2": {}</pre>
<b>OperatorDivision</b>	<ul style="list-style-type: none"><li>• Allows to divide <b>expression1</b> by <b>expression2</b>.</li></ul>	<pre>"type": "OperatorDivision", "expression1": {}, "expression2": {}</pre>
<b>OperatorEquals</b>	<ul style="list-style-type: none"><li>• Allows the comparison of <b>expression1</b> and <b>expression2</b>.</li></ul>	<pre>"type": "OperatorEquals", "expression1": {}, "expression2": {}</pre>

### On this Page:

- [Available Operators](#)
- [Available Expressions](#)
- [Calculation Rule Examples](#)
  - [Calculating the Duration Between two Steps](#)
  - [Adding two Values](#)
  - [Subtracting two Values](#)

### Related Pages:

- [Using Calculated Metrics](#)
  - [Adding Calculated Metrics](#)
  - [Editing Calculated Metrics](#)

<b>OperatorIndex</b>	<ul style="list-style-type: none"> <li>Allows to extract the process node at a given index (second operand) from a set of process nodes. This set of process nodes is itself the result of an expression (first operand).</li> </ul>	<pre>"type": "OperatorIndex", "expression1": {}, "expression2": {}</pre>
<b>OperatorMultiplication</b>	<ul style="list-style-type: none"> <li>Allows to multiply <b>expression1</b> and <b>expression2</b>.</li> </ul>	<pre>"type": "OperatorMultiplication", "expression1": {}, "expression2": {}</pre>
<b>OperatorProcessAttribute</b>	<ul style="list-style-type: none"> <li>Allows to extract custom attributes from the given process.</li> </ul>	<pre>"type": "OperatorProcessAttribute", "expression": {}</pre>
<b>OperatorSubtraction</b>	<ul style="list-style-type: none"> <li>Allows to subtract <b>expression2</b> from <b>expression1</b>.</li> </ul>	<pre>"type": "OperatorSubtraction", "expression1": {}, "expression2": {}</pre>

## Available Expressions

The following table of expressions can be searched. To do this, enter your search phrase in the search field:

Expression	Description	Structure
<b>FilterAttribute</b>	<ul style="list-style-type: none"> <li>Allows to extract an attribute from a process node. Only the first process node in the evaluation context is considered.</li> <li>This filter can be used in a <b>LocationStep</b>.</li> </ul>	<pre>"type": "FilterAttribute", "expression": {}</pre>
<b>FilterIndex</b>	<ul style="list-style-type: none"> <li>Allows to extract the process node at the given index from the evaluation context.</li> <li>This filter can be used in a <b>LocationStep</b>.</li> </ul>	<pre>"type": "FilterIndex", "expression": {}</pre>

<b>FilterType</b>	<ul style="list-style-type: none"> <li>Allows to extract the type from a process node. Only the first process node in the evaluation context is considered.</li> <li>This filter can be used in a <b>LocationStep</b>.</li> </ul>	<pre>"type": "FilterT ype"</pre>
<b>LocationPath</b>	<ul style="list-style-type: none"> <li>A <b>LocationPath</b> is a set of <b>LocationSteps</b> to be applied on the evaluation context.</li> </ul>	<pre>"type": "Locatio nPath", "steps": [ ]</pre>
<b>LocationStep</b>	<ul style="list-style-type: none"> <li>Represents a single step in a <b>LocationPath</b>. <ul style="list-style-type: none"> <li><b>axis</b>: The given axis defines how to traverse the evaluation context. It indicates the direction of travel (possible values: descendant, ancestor).</li> <li><b>nodeTest</b>: Defines the name of the nodes to select on the given axis.</li> <li><b>filters</b> (optional): Allows to filter the selected process nodes.</li> </ul> </li> </ul>	<pre>"type": "Locatio nStep", "axis": &lt;&gt;, "nodeTes t": &lt;&gt;, "filters" : &lt;&gt;</pre>
<b>ValueBoolean</b>	<ul style="list-style-type: none"> <li>Represents a boolean value or operand for operating expressions.</li> </ul>	<pre>"type": "ValueBo olean", "value": &lt;&gt;</pre>
<b>ValueNumber</b>	<ul style="list-style-type: none"> <li>Represents a number value or operand for operating expressions.</li> </ul>	<pre>"type": "ValueNu mber", "value": &lt;&gt;</pre>
<b>ValueString</b>	<ul style="list-style-type: none"> <li>Represents a string value or operand for operating expressions.</li> </ul>	<pre>"type": "ValueSt ring", "value": &lt;&gt;</pre>

## Calculation Rule Examples



Currently it is only possible to offset **numbers** against each other.

### Calculating the Duration Between two Steps

<b>Use Case</b>	You want to calculate the duration between <b>step1</b> and <b>step2</b> . Each step is passed only once.
-----------------	---

<b>Description</b>	<ul style="list-style-type: none"> <li>• To subtract two values, you need the <b>OperatorSubtraction</b>. The operator needs <b>expression1</b> and <b>expression2</b>.</li> <li>• For each expression you have to extract the correct attribute/value from a process node. To do so, you need the <b>OperatorAttribute</b>. <b>expression1</b> extracts the correct step and <b>expression2</b> gets the correct value from this step.</li> <li>• Then you need the <b>LocationPath</b> expression. Fill the <b>steps</b> array with the process step you need. Add this step to the property <b>nodeTest</b> . You also have to indicate the direction, <b>descendant</b> would be the default.</li> </ul>
<b>Example Calculation Rule</b>	<pre> {   "type": "OperatorSubtraction",   "expression1": {     "type": "OperatorAttribute",     "expression1": {       "type": "LocationPath",       "steps": [{         "type": "LocationStep",         "axis": "descendant",         "nodeTest": "step2"       }]     },     "expression2": {       "type": "ValueString",       "value": "start"     }   },   "expression2": {     "type": "OperatorAttribute",     "expression1": {       "type": "LocationPath",       "steps": [{         "type": "LocationStep",         "axis": "descendant",         "nodeTest": "step1"       }]     },     "expression2": {       "type": "ValueString",       "value": "end"     }   } } </pre>

## Adding two Values

<b>Use Case</b>	You want to add <b>CustomAttribute1</b> and <b>CustomAttribute2</b> .
<b>Description</b>	<ul style="list-style-type: none"> <li>• To add two attributes, you need the <b>OperatorAddition</b>. This operator needs two expressions.</li> <li>• Each expression should contain the value of one custom attribute. To get the values, you need the <b>OperatorProcessAttribute</b>. Apply this for <b>expression1</b> and <b>expression2</b>.</li> <li>• In our example, the custom attributes are strings - which must be indicated. Therefore, we need to use expression <b>ValueString</b> (even if the actual value used for the calculation is a number).</li> </ul>

<b>Example Calculation Rule</b>	<pre> {   "type": "OperatorAddition",   "expression1": {     "type": "OperatorProcessAttribute",     "expression": {       "type": "ValueString",       "value": "CustomAttribute1"     }   },   "expression2": {     "type": "OperatorProcessAttribute",     "expression": {       "type": "ValueString",       "value": "CustomAttribute2"     }   } } </pre>
---------------------------------	---

## Subtracting two Values

<b>Use Case</b>	You want to subtract <b>CustomAttribute2</b> from <b>CustomAttribute1</b> .
<b>Description</b>	<ul style="list-style-type: none"> <li>• To subtract two attributes, you need the <b>OperatorSubtraction</b>. This operator needs two expressions.</li> <li>• Each expression should contain the value of one custom attribute. To get the values, you need the <b>OperatorProcessAttribute</b>. Apply this for <b>expression1</b> and <b>expression2</b>.</li> <li>• In our example, the custom values are strings - which must be indicated. Therefore, we need to use expression <b>ValueString</b> (even if the actual value used for the calculation is a number).</li> </ul>
<b>Example Calculation Rule</b>	<pre> {   "type": "OperatorSubtraction",   "expression1": {     "type": "OperatorProcessAttribute",     "expression": {       "type": "ValueString",       "value": "CustomValue1"     }   },   "expression2": {     "type": "OperatorProcessAttribute",     "expression": {       "type": "ValueString",       "value": "CustomValue2"     }   } } </pre>