

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.

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](#)

Available Operators

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

Operator	Description	Structure
OperatorAddition	<ul style="list-style-type: none">• Allows the addition of expression2 and expression1.	<pre>"type": "OperatorAddition", "expression1": {}, "expression2": {}</pre>
OperatorAttribute	<ul style="list-style-type: none">• 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).• Possible values for expression2: start, end, name (of a process step).	<pre>"type": "OperatorAttribute", "expression1": {}, "expression2": {}</pre>
OperatorDivision	<ul style="list-style-type: none">• Allows to divide expression1 by expression2.	<pre>"type": "OperatorDivision", "expression1": {}, "expression2": {}</pre>
OperatorEquals	<ul style="list-style-type: none">• Allows the comparison of expression1 and expression2.	<pre>"type": "OperatorEquals", "expression1": {}, "expression2": {}</pre>

OperatorIndex	<ul style="list-style-type: none"> 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). 	<pre>"type": "OperatorIndex", "expression1": {}, "expression2": {}</pre>
OperatorMultiplication	<ul style="list-style-type: none"> Allows to multiply expression1 and expression2. 	<pre>"type": "OperatorMultiplication", "expression1": {}, "expression2": {}</pre>
OperatorProcessAttribute	<ul style="list-style-type: none"> Allows to extract custom attributes from the given process. 	<pre>"type": "OperatorProcessAttribute", "expression": {}</pre>
OperatorSubtraction	<ul style="list-style-type: none"> Allows to subtract expression2 from expression1. 	<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
FilterAttribute	<ul style="list-style-type: none"> Allows to extract an attribute from a process node. Only the first process node in the evaluation context is considered. This filter can be used in a LocationStep. 	<pre>"type": "FilterAttribute", "expression": {}</pre>
FilterIndex	<ul style="list-style-type: none"> Allows to extract the process node at the given index from the evaluation context. This filter can be used in a LocationStep. 	<pre>"type": "FilterIndex", "expression": {}</pre>

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

Calculation Rule Examples



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

Calculating the Duration Between two Steps

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

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

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

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

Subtracting two Values

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