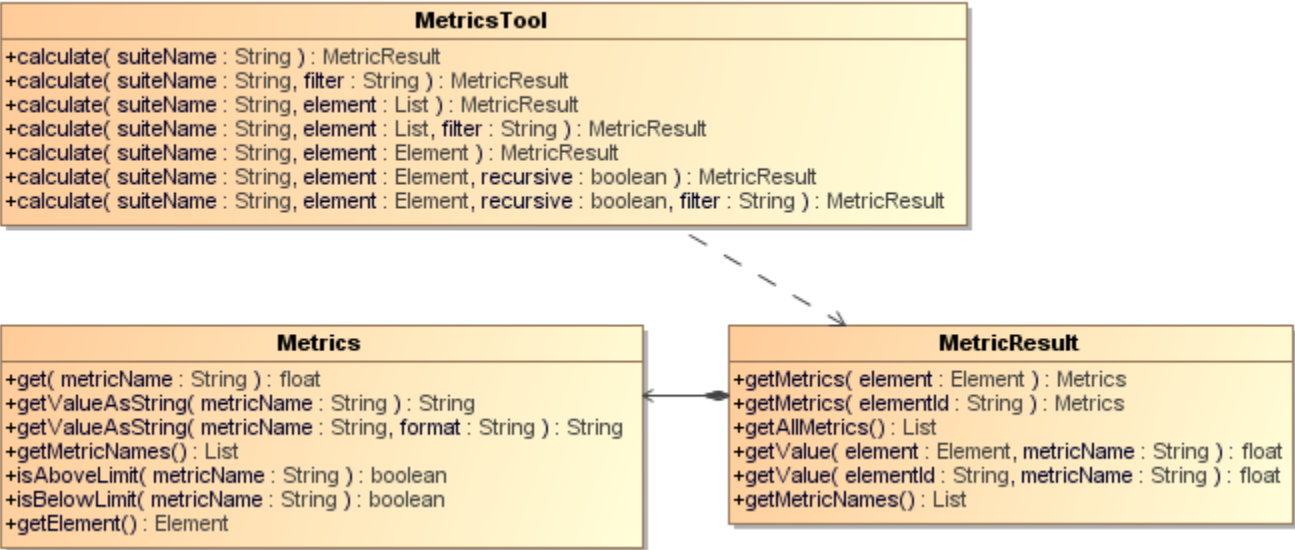


Metrics tool API

On this page

- [Calculation Methods](#)
- [Getting Metric Data from Metric Results](#)
- [Getting Metric Values](#)
- [Getting the Metric Name List](#)
- [Getting the Result Attribute "is above limit" from a Metrics Name](#)
- [Getting the Result Attribute "is below limit" from a Metric Name](#)



A Class diagram of the MetricsTool.

Class of MetricsTool

Methods:

- +calculate(suiteName : String) : MetricsResult
- +calculate(suiteName : String, filter : String) : MetricsResult
- +calculate(suiteName : String, element : java::util::List) : MetricsResult
- +calculate(suiteName : String, element : java::util::List, filter : String) : MetricsResult
- +calculate(suiteName : String, element : Element) : MetricsResult
- +calculate(suiteName : String, element : Element, recursive : boolean) : MetricsResult
- +calculate(suiteName : String, element : Element, filter : String) : MetricsResult

Class of MetricsResult

Methods:

- +getMetrics(element : Element) : Metrics
- +getMetrics(elementId : String) : Metrics
- +getAllMetrics() : List<Metrics>
- +getValue(element : Element, metricName : String) : float
- +getValue(elementId : String, metricName : String) : float
- +getMetricNames() : List<String>

Class of Metric

Methods:

- +get(metricName : String) : float
- +getValueAsString(metricName : String) : String
- +getValueAsString(metricName : String, format : String) : String
- +getMetricNames() : List
- +isAboveLimit(metricName : String) : boolean
- +isBelowLimit(metricName : String) : boolean

Calculation Methods

calculate(suiteName : String) : MetricResult

Uses only the suite name to calculate the Metric from an entire project.

```
#set ($metricsResults = $metrics.calculate('suiteName'))
```

Example code:

Calculates the Metric from the Actor name "Template Author".

```
#set ($metricsResults = $metrics.calculate('System Metrics'))
```

calculate(suiteName : String, filter : String) : MetricResult

Uses a suite name and filter to calculate the Metric from an entire project. "filter" is a filter name displayed in the Metric dialog.

```
#set ($metricsResults = $metrics.calculate('suiteName', "filter"))
```



If the language used in the MagicDraw interface changes, the filter name needs to be replaced with a target language, for example: French will be "Violations de Paquetage" for "Package Violations".

calculate(suiteName : String, elementList : List<Element>) : MetricResult

Specifies a suite name and Element list to calculate the Metrics.

```
#set ($metricsResults = $metrics.calculate('suiteName', $elementList))
```

To calculate the Metrics from all classes by using the 'System Metrics' suite, for example, type the following code:

```
#set ($metricsResults = $metrics.calculate('System Metrics', $Class))
```

calculate(suiteName : String, elementList : List<Element>, filter : String) : MetricResult

Specifies a suite name and Element list to calculate the Metric with a specific filter name.

```
#set ($metricsResults = $metrics.calculate('suiteName', $elementList, "filter"))
```

To calculate the Metric from all classes by using the 'System Metrics' suite with the package violations filter, for example, type the following code:

```
#set ($metricsResults = $metrics.calculate('System Metrics', $Class, "Package Violations"))
```

calculate(suiteName : String, element : Element) : MetricResult

Calculates the Metric from a single element.

```
#set ($metricsResults = $metrics.calculate('suiteName', $element))
```

To calculate the Metric from the Actor name "Template Author", for example, type the following code:

```
#set ($actor = $report.findElementInCollection($Actor, "Template Author"))  
#set ($metricsResults = $metrics.calculate('System Metrics', $actor))
```

calculate(suiteName : String, element : Element, recursive : boolean) : MetricResult

Calculates the Metric from a single element with the recursive option. If the recursive option is true, it will calculate the result from a given root element and recursive to all children; otherwise (false), calculates the result from the single element.

```
#set ($metricsResults = $metrics.calculate('suiteName', $element, $recursive))
```

To calculate the Metric from a root element and recursive to all children. The result is similar to the calculated result from all elements in a project, for example, type the following code:

```
#set ($root = $project.model)
#set ($metricsResults = $metrics.calculate('System Metrics', $root, true))
```

calculate(suiteName : String, element : Element, recursive : Boolean, filter : String) : MetricResult

Calculates the Metric from a single element with the recursive option and a filter name. If the recursive option is true, it will calculate the result from a given root element and recursive to all children; otherwise (false), calculate the result from the single element.

```
#set ($metricsResults = $metrics.calculate('suiteName', $element, $recursive, "filter"))
```

Getting Metric Data from Metric Results

The object returned from calculating methods in the previous section is the Metrics results. The Metrics results represent the Metrics Table as shown in MagicDraw. It consists of Element and the Metrics. To get the Metrics and value, we need to pass the element (row) as parameter and get the Metrics (column).

getMetrics(element : Element) : Metrics

Gets the Metric of a specified element.

```
#set ($elementMetric = $metricsResults.getMetrics($element))
```

getMetrics(elementId : String) : Metrics

Gets the Metric of a specific element ID.

```
#set ($elementMetric = $metricsResults.getMetrics("element id"))
```

getAllMetrics() : List<Metrics>

Gets all Metrics at once.

To print all numbers of attribute (NA) value of all elements, for example, type the following code:

```
#set ($metricsResults = $metrics.calculate('System Metrics'))
#foreach ($elementMetric in $metricsResults.allMetrics)
$elementMetric.element.name has $elementMetric.NA
#end
```

getValue(element : Element, metricName : String) : float

This is the shortcut method to get a specific Metric value of a target element. The Metric name is the abbreviation of Metric, for example, NA, NC, NO, RFC, etc. This value is case-sensitive. See section **Getting Metric Values**.

```
#set ($value = $metricsResults.getValue($element, "Metric name"))
```

getValue(elementId : String, metricName : String) : float

This is the shortcut method to get the specified Metrics value of a target element. The Metric name is the abbreviation of Metric, for example, NA, NC, NO, RFC, etc. This value is case-sensitive. See 21.1.3 Getting Metric Values.

```
#set ($value = $metricsResults.getValue("element id", "Metric name"))
```

Getting Metric Values

The method in Section **Getting Metric Data from Metric Results** will return a Metric object. The Metric object represents all Metric values of a single element. If the Metric contains no value, the returned value will be zero.

get(metricName : String) : float

The Metric name is the abbreviation of Metric, for example, NA, NC, NO, and RFC. This value is case-sensitive. To return the number of attributes (NA) from Class SwingUtilities, for example, type the following code:

```
#set ($class = $report.findElementInCollection($Class, "SwingUtilities"))
#set ($metricsResults = $metrics.calculate('System Metrics', $class))
#set ($elementMetric = $metricsResults.getMetrics($class))
#set ($value = $elementMetric.get("NA"))
```

The user can use the Velocity technique for a shortcut when retrieving value from a Metrics name. The Metrics name is case-sensitive.

To return the number of an attribute (NA), for example, type the following code:

```
#set ($value = $elementMetric.NA)
```

getValueAsString(metricName : String) : String

You can format an output value as string. The Metric name is the abbreviation of Metric, for example, NA, NC, NO, and RFC. This value is case-sensitive. If the Metric contains no value, the returned string will be an empty string.

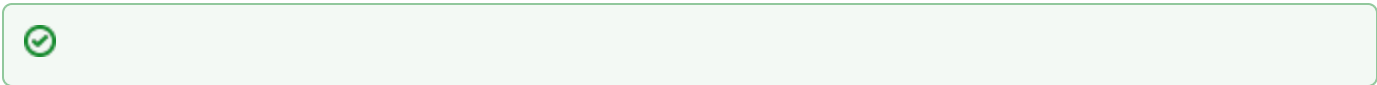
```
#set ($value = $elementMetric.getValueAsString('metricName'))
```

The characters in the table below are used in the patterns.

| Symbol | Meaning |
|--------|---|
| 0 | Digit |
| # | Digit, zero shows as absent |
| . | Decimal separator or monetary decimal separator |
| , | Grouping separator |
| E | Separates mantissa and exponent in scientific notation. |
| % | Multiply by 100 and show as percentage |

The following are some examples of patterns.

| Value | Pattern | Result |
|----------|------------|---------------|
| 23.456 | ###.00 | 23.46 |
| 23.456 | 000.00 | 023.46 |
| 23.456 | ###.### | 23.456 |
| 12345678 | ###,###.## | 12,345,678.00 |
| 12345678 | #.## | 12345678 |



The following is the sample code:

Return the number of an attribute (NA) from Class SwingUtilities.

```
#set ($class = $project.findElementCollection($class, 'SwingUIElements'))
#set ($metricsResults = $metrics.calculate('System Metrics', $class))
#set ($elementMetric = $metricsResults.getMetrics($class))
#set ($value = $elementMetric.getValueAsString("NA", '#.##'))
```

\$elementMetric.getValueAsString("NA", "#.##")

Getting the Metric Name List

The code written above will return an unexpected result. Use the following code to get the formatted text.

A template author can list an entire available Metric names from 'metricNames' methods. The 'metricNames' is available in both Metric results and Metric class.

However, this method cannot use a single quote character in formatted text.

getMetricNames() : List<String>

```
#set ($nameList = $elementMetric.metricNames)
```

Or

```
#set ($nameList = $metricsResults.metricNames)
```

\$elementMetric represents a Metric class and \$metricsResults represents a MetricsResults class.

To print all Metric names and all of their classes' values, for example, type the following code:

```
#set ($metricsResults = $metrics.calculate('System Metrics', $Class))
#foreach ($class in $Class)
$class.humanName
#foreach ($metricName in $metricsResults.metricNames)
$metricName value $metricsResults.getValue($class, $metricName)
#end
#end
```

To print all Metric names and their values of all Classes, for example, type the following code:

```
#set ($metricsResults = $metrics.calculate('System Metrics', $Class))
#foreach ($class in $Class)
#foreach ($elementMetric in $metricsResults.getMetrics($class))
#foreach ($metricName in $elementMetric.metricNames)
$metricName value $elementMetric.get($metricName)
#end
#end
#end
```

Getting the Result Attribute “is above limit” from a Metrics Name

The “is above limit” is a Boolean value that indicates that a value is larger than the value configured in the Metrics menu. The \$elementMetric represents a Metric class and \$metricsResults represents a MetricsResults class.

```
#set ($isAbove = $metricsResults.isAboveLimit('metricName'))
```

To change the HTML text color to red if the number of classes in a project is larger than the limit, for example, type the following code:

```
#set ($metricsResults = $metrics.calculate('System Metrics', $project.model, true))
#set ($elementMetrics = $metricsResults.getMetrics($project.model))
#if ($elementMetrics.isAboveLimit('NC'))
Total classes is <font color="red">$elementMetrics.get('NC')</font>
#end
#end
```

Getting the Result Attribute “is below limit” from a Metric Name

The “is below limit” is a Boolean value that indicates that a value is less than the value configured in the Metric menu.

The \$elementMetric represents a Metric class.

```
#set ($isAbove = $elementMetric.isBelowLimit('metricName'))
```

Example code:

Change the text color to red if the number of responses for the class “Customer” is less than the limit.

```
#set ($class = $report.findElementInCollection($Class, "Customer"))
#set ($metricsResults = $metrics.calculate('System Metrics', $class))
#set ($elementMetrics = $metricsResults.getMetrics($class))
#if ($elementMetrics.isBelowLimit('RFC'))
$class.name is too complex. The total number of methods access to this class is <font color="blue"
>$elementMetrics.get('RFC')</font>
#end
```