

Miscellaneous Asserts

Last Modified on 15 February 2022

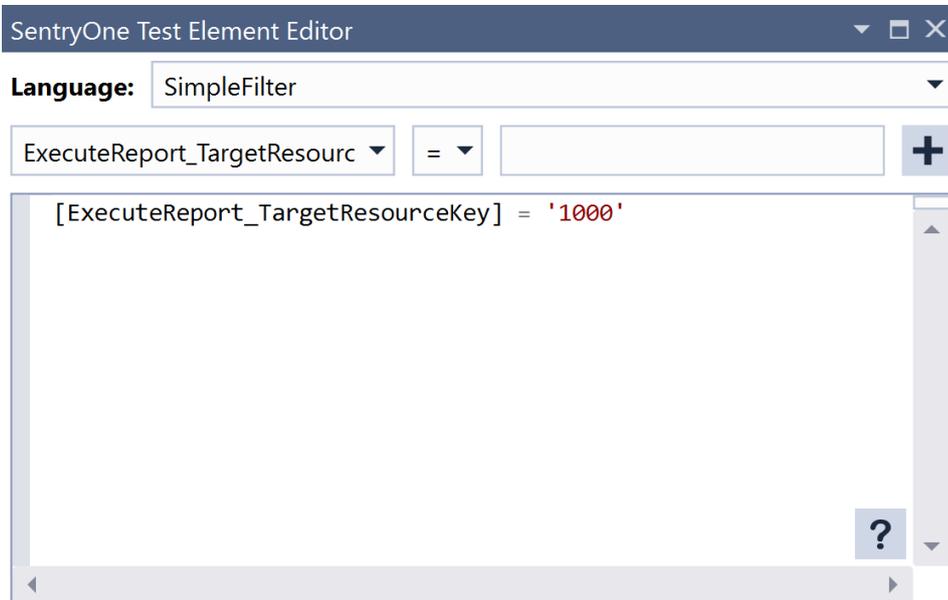
EOL: SentryOne Test will reach its [end of life](#) date on June 15, 2022. See the [Solarwinds End of Life Policy](#) for more information.

The following list is a collection of asserts that can compare a variety of object types.

Expression

Icon	Description
	The Expression assert lets you enter a free-form expression to use more powerful combinational logic to validate that one or more resource values are correct.

Expression Editor



Function	Description
Language	Select the expression language. Currently only the SimpleFilter language is available.
	To add a clause to the expression, simply select the resource in the left hand combo box, choose the operator in the middle combo box and type the value you require in

Function Filter Expression	<p>the right hand text box. Selecting the + button adds the specified clause to the Description expression. For more information about filter expressions, please see the filtering topic.</p> <p>Note: Help on the filtering language is also available in the editor directly by selecting on the ? icon in the bottom right.</p>
-----------------------------------	---

Often when filtering, whether in SQL or using SentryOne Test filters, the identifiers relate to field names. In this instance, the identifiers relate to resource keys. For example, if there was an **Execute Query Scalar** action that generated a resource called **MyScalarValue** then the expression could include a clause such as **[MyScalarValue] > 10**.

Integer Comparison

Icon	Description
	<p>The Integer Comparison assert tests two integers for different levels of equality. Specify to test for greater than, less than, or even user defined tolerances. The integer to compare your actual value to can either be a static value defined in the element's editor, or from another resource key.</p>

Important: The Integer Comparison assertion is now deprecated. While existing projects can still continue to use it, new products should migrate to using the Value Comparison assertion.

Integer Comparison Editor

Integer comparison

Comparable resource key	
Comparable value source	StaticValue
Comparison mode	Equal
Comparison value	2
Resource key	1 (Scalar Value)
Tolerance	0

Comparable value source
Whether to use a static value or resource to compare against.

Function	Description														
Comparable Resource Key	Select a resource key you want to compare against if you opt not to use a static value to compare.														
Comparable Value Source	Specifies if the element should use a static value or a resource key to make the comparison.														
Comparison Mode	<table border="1"> <thead> <tr> <th>Mode</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Equal</td> <td>Asserts that the two integers are equal.</td> </tr> <tr> <td>NotEqual</td> <td>Asserts that the two integers are not equal.</td> </tr> <tr> <td>LessThan</td> <td>Asserts that the integer is less than the static value or resource.</td> </tr> <tr> <td>GreaterThan</td> <td>Asserts that the integer is greater than the static value or resource.</td> </tr> <tr> <td>Within Percentage Tolerance</td> <td>Asserts that the integer is within a specified percent of the static value or resource.</td> </tr> <tr> <td>Within Fixed Tolerance</td> <td>Asserts that the integer is within the specified tolerance range of the static value or resource.</td> </tr> </tbody> </table>	Mode	Description	Equal	Asserts that the two integers are equal.	NotEqual	Asserts that the two integers are not equal.	LessThan	Asserts that the integer is less than the static value or resource.	GreaterThan	Asserts that the integer is greater than the static value or resource.	Within Percentage Tolerance	Asserts that the integer is within a specified percent of the static value or resource.	Within Fixed Tolerance	Asserts that the integer is within the specified tolerance range of the static value or resource.
Mode	Description														
Equal	Asserts that the two integers are equal.														
NotEqual	Asserts that the two integers are not equal.														
LessThan	Asserts that the integer is less than the static value or resource.														
GreaterThan	Asserts that the integer is greater than the static value or resource.														
Within Percentage Tolerance	Asserts that the integer is within a specified percent of the static value or resource.														
Within Fixed Tolerance	Asserts that the integer is within the specified tolerance range of the static value or resource.														
Comparison Value	The static value to use if you opt for a static value over a resource.														

Resource Key	The resource key of the integer you want to compare.
Tolerance	Specifies the fixed or percent tolerance.

Range Check

Icon	Description
	The Range Check assert checks a resource key's value to see if it's between the user set maximum and minimum value.

Range Check Editor

Properties
⌵ 🔍 ✕

Assertion Range Check

🔧 ⚙️ 🔑

Misc

IsEnabled	True
Maximum Value	0
Minimum Value	0
Name	Range Check
Uniqueld	42c3e2f3-a73c-4143-b8a6-1f103375a110
Value Resource Key	No resource selected

Function	Description
Maximum Value	The maximum allowable value for the Value Resource Key.
Minimum Value	The minimum allowable value for the Value Resource Key.
Value Resource Key	Selects a resource key where you want to compare a value.

Scalar Comparison

Icon	Description



The **Scalar Comparison** assert takes the result of an **Execute Scalar** action and compares it with either a static value or another scalar result. The scalar values can be several different types, each with their own specifications of equality.

⚠ Important: The **Scalar Comparison** assertion is now deprecated. While existing projects can still continue to use it, new products should migrate to using the Value Comparison assertion.

Scalar Comparison Editor

LegiTest Element Editor

^ **Scalar comparison**

Comparable resource key	
Comparable value source	StaticValue
Comparison type	Boolean
Comparison value	True
Equality type	Equal
Resource key	1 (Scalar Value)
Tolerance	0

Comparable value source
Whether to use a static value or resource to compare against.

Function	Description								
Comparable Resource Key	Enter the resource key for the dynamic value if you do not opt to use a static value to compare against.								
Comparable Value Source	Specify whether to use a static value or a dynamic value from another resource key.								
Comparison Type	Specify the type of the value to compare: <table border="1" style="margin-left: 20px;"> <tr> <td>• Boolean</td> <td>• Decimal</td> <td>• Guid</td> <td>• Text</td> </tr> <tr> <td>• DateTime</td> <td>• Floating Point</td> <td>• Integer</td> <td></td> </tr> </table>	• Boolean	• Decimal	• Guid	• Text	• DateTime	• Floating Point	• Integer	
• Boolean	• Decimal	• Guid	• Text						
• DateTime	• Floating Point	• Integer							
	Specify the equality type:								

Function Equality Type	Description	<ul style="list-style-type: none"> • Equal • Within Percent Tolerance
		<ul style="list-style-type: none"> • Not Equal • Within Fixed Tolerance
Resource Key	The value to compare.	
Tolerance	Specify fixed or percent tolerance for certain value types.	

Value Comparison

Icon	Description
	The Value Comparison assert is a general purpose comparison for comparing two single values. It supersedes both the Integer Comparison and the Scalar Comparison assertions, presenting a simpler configuration while also significantly extending the available options.

Value Comparison Editor

SentryOne Test Element Editor

Actual value:
Resource:

Expected value:
 Static value Resource
 Static value:

Comparison:
 Data type:
 Comparison type:

Summary:
 Validate that test resource 'ExecuteReport_TargetResourceKey' will be exactly equal to "

Static Value

SentryOne Test Element Editor

Actual value:
Resource:

Expected value:
 Static value Resource
 Resource:

Comparison:
 Data type:
 Comparison type:

Summary:
 Validate that test resource 'ExecuteReport_TargetResourceKey' will be exactly equal to test resource "

Resource

There are four main sections to the editor:

Function	Description
Actual Value	This is the value that you want to validate. It always comes from a resource that's available in the test. To select the resource to test, select the arrow to the right of

Function	the resource name box. Description						
Expected Value	This is the value that you expect the actual value to match. This can either come from a static value that you configure in the user interface or from a resource. If you're using a static value, type the value that you want into the static value box. If you're using a resource then select that option and pick the resource in the same way you did for the actual value.						
Comparison	<p>This section specifies the options that we use for comparison:</p> <table border="1"> <thead> <tr> <th>Comparison</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Data type</td> <td>Detected automatically when the actual and expected values are entered, but can be changed manually to use a different type for the comparison if necessary.</td> </tr> <tr> <td>Comparison type</td> <td> <p>This is the type of comparison that you want to use, and different types are available for different comparison types.</p> <p>For example, 'GreaterThan' is not available for boolean comparisons and 'StringMatch' is not available for integers. Some comparison types have additional configuration options, which are detailed below.</p> <p>For comparison types 'WithinFixedTolerance' and 'WithinPercentageTolerance' the tolerance options are visible. There is the choice of entering a symmetric tolerance (i.e. same amount of tolerance up and down) or the values for the upper and lower tolerance can be entered specifically. By way of example, in the image above where we are validating within a fixed tolerance of 11 with 5 down and 10 up, values between 6 (11-5) and 21 (11+10) would pass the check.</p> <p>For comparison types 'StringMatch' and 'RegexMatch', additional string comparison options are available - including whether to ignore case, ignore line endings and trim spaces from the start / end of the values to compare. An example of this is shown in the image below.</p> </td> </tr> </tbody> </table>	Comparison	Description	Data type	Detected automatically when the actual and expected values are entered, but can be changed manually to use a different type for the comparison if necessary.	Comparison type	<p>This is the type of comparison that you want to use, and different types are available for different comparison types.</p> <p>For example, 'GreaterThan' is not available for boolean comparisons and 'StringMatch' is not available for integers. Some comparison types have additional configuration options, which are detailed below.</p> <p>For comparison types 'WithinFixedTolerance' and 'WithinPercentageTolerance' the tolerance options are visible. There is the choice of entering a symmetric tolerance (i.e. same amount of tolerance up and down) or the values for the upper and lower tolerance can be entered specifically. By way of example, in the image above where we are validating within a fixed tolerance of 11 with 5 down and 10 up, values between 6 (11-5) and 21 (11+10) would pass the check.</p> <p>For comparison types 'StringMatch' and 'RegexMatch', additional string comparison options are available - including whether to ignore case, ignore line endings and trim spaces from the start / end of the values to compare. An example of this is shown in the image below.</p>
Comparison	Description						
Data type	Detected automatically when the actual and expected values are entered, but can be changed manually to use a different type for the comparison if necessary.						
Comparison type	<p>This is the type of comparison that you want to use, and different types are available for different comparison types.</p> <p>For example, 'GreaterThan' is not available for boolean comparisons and 'StringMatch' is not available for integers. Some comparison types have additional configuration options, which are detailed below.</p> <p>For comparison types 'WithinFixedTolerance' and 'WithinPercentageTolerance' the tolerance options are visible. There is the choice of entering a symmetric tolerance (i.e. same amount of tolerance up and down) or the values for the upper and lower tolerance can be entered specifically. By way of example, in the image above where we are validating within a fixed tolerance of 11 with 5 down and 10 up, values between 6 (11-5) and 21 (11+10) would pass the check.</p> <p>For comparison types 'StringMatch' and 'RegexMatch', additional string comparison options are available - including whether to ignore case, ignore line endings and trim spaces from the start / end of the values to compare. An example of this is shown in the image below.</p>						
Summary	Provides a plain text description of what's configured in the assertion editor.						

Xml Comparison

Icon	Description
	The Xml Comparison assert compares two Xml resources. These resources

Icon



can be loaded many different ways, such as from a **Load Asset** that's **Description** the **Content Asset**, result of a query, or even from a **Get Properties** Action. The Xml in the resource passes in as a string, and checks if it's valid Xml compared to the other resource.

Xml Comparison Editor

Properties

Assertion Xml Comparison

Misc

Actual Xml Resource Key	ExecuteReport_TargetResourceKey (XML String)
Actual XPath Comparison	
Actual XPath Exclusion	
Compare Attribute Order	False
Expected Xml Resource Key	ExecuteReport_TargetResourceKey (XML String)
Expected XPath Comparison	
Expected XPath Exclusion	
Ignore White Space	False
IsEnabled	True
Name	Xml Comparison
Uniqueld	fb4e954c-ab75-4741-9953-f5be4a5307f8

Function	Description
Actual Xml Resource Key	The resource key for the Xml you want to compare.
Compare Attribute Order	When enabled, the comparison compares the order of attributes in an element, before comparing the content. If left disabled, only the attribute value is compared.
Expected Xml Resource Key	The resource key for the Xml you want to compare the Actual Xml against.
Ignore White Space	Specifies whether to ignore any extra spaces or white spaces in or around elements.
Actual XPath Exclusion	This field takes an XPath string that executes against the Actual Xml once it loads. Elements found by the XPath are excluded from the comparison on the Actual side only. This field can be a (pipe) delimited string, allowing for multiple XPath items to execute.
Expected XPath Exclusion	This field takes an XPath string that executes against the Expected Xml once it loads. Elements found by the XPath are excluded from the comparison on the Expected side only. This field can be a (pipe) delimited string, allowing for multiple XPath items to execute.
Actual XPath Comparison	This field takes an XPath string that executes against the Actual Xml once it loads. Only the elements found by the XPath are used in the comparison for the Actual side only.

Function Expected XPath Comparison	Description This field takes an XPath string that executes against the Expected Xml once it loads. Only the elements found by the XPath are used in the comparison for the Expected side only.
---	---