



## Description of the method

<b>Description:</b>	<p>= and &lt;&gt; operators compare two values and are available for all types of elements</p> <p>= returns true if values are equal</p> <p>&lt;&gt; returns true if values are not equal</p> <p>Rest of the operators (&gt;=, &lt;=, &lt;, &gt;) are available only for double/decimal and integer type of elements.</p> <p>&gt;= returns true if left side operand is greater or equal to right side operand</p> <p>&lt;= returns true if left side operand is less or equal to right side operand</p> <p>&gt; returns true if left side operand is greater than right side operand</p> <p>&lt; returns true if left side operand is less than right side operand</p>
<b>Available for:</b>	<p>= and &lt;&gt;: base64binary, boolean, datetime, double/decimal, date, hexBinary, integer, string</p> <p>&gt;=, &lt;=, &lt;, &gt;: double/decimal and integer</p>
<b>Parameters:</b>	One parameter for comparison. Both operands of the comparison must have same type.
<b>Return type:</b>	boolean

### Example1

<b>Context:</b>	HeaderType1
<b>OCL:</b>	self.Id = "IdContent"
<b>Description:</b>	The example rule compares value of <Id> element to a string "IdContent".

The XML snippet below would pass this check.

```
<Id>IdContent</Id>
```

The other snippet below however would not pass this check as the value is different from "IdContent".

```
<Id>Id Content</Id>
```

## Example2

<b>Context:</b>	HeaderType1
<b>OCL:</b>	self.Id <> "IdContent"
<b>Description:</b>	The example rule compares value of <Id> element to a string "IdContent". The value is expected to be different from the string, otherwise an error will be returned by the rule.

The XML snippet below would pass this check.

```
<Id>SomeOtherContent</Id>
```

The other snippet below however would not pass this check as the value is equal to "IdContent".

```
<Id>IdContent</Id>
```

Attached you can find example schema and files which you can use for testing.