



## Description of the method

<b>Description:</b>	<p>Logical operators compare expressions and return a boolean value, depending on which operator is used</p> <p><i>expr1</i> and <i>expr2</i>: Returns true, when both expressions are true</p> <p><i>expr1</i> or <i>expr2</i>: Returns true, when at least one expression is true</p> <p><i>expr1</i> xor <i>expr2</i>: Returns true, when exactly one expression is true</p> <p><i>not(expr)</i>: Returns the opposite of whichever <i>expr</i> is</p>
<b>Available for:</b>	base64binary, boolean, datetime, double/decimal, date, hexBinary, integer, string, double/decimal and integer
<b>Parameters:</b>	-
<b>Return type:</b>	boolean

### Example1

<b>Context:</b>	TransactionType1
<b>OCL:</b>	<pre>self.Amount &gt; 200 implies self.Debtor.Name-&gt;size() = 1 and self.Creditor.Name-&gt;size() = 1</pre>
<b>Description:</b>	The example rule mandates the usage of both Creditor and Debtor name when Amount exceeds 200

The XML snippet below would pass this check.

```

<Transaction>
  <Id>203</Id>
  <Amount>201</Amount>
  <Debtor>
    <Name>Debtor2</Name>
  </Debtor>
  <Creditor>
    <Name>Creditor2</Name>
  </Creditor>
</Transaction>
<Transaction>
  <Id>203</Id>
  <Amount>1</Amount>
  <Creditor>
    <Name>Creditor2</Name>
  </Creditor>
</Transaction>
<Transaction>

```

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

```

<Transaction>
  <Id>203</Id>
  <Amount>201</Amount>
  <Creditor>
    <Name>Creditor2</Name>
  </Creditor>
</Transaction>
<Transaction>

```

Please note that in order to make the feedback for user as accurate as possible, this exact rule may be divided into two separate rules. The individual rules would be:

if Amt exceeds 200, then Creditor Name is mandatory

if Amt exceeds 200, then Debtor Name is mandatory

This way error report will always point out the erroneous case, as the query and error message for each rule can be different.