



6.7 Rule Expressions

Previously, we gave some basic examples on how RuleExpressions should be written. This section provides more detailed information and examples on how to design expressions and use rule expression operators with each data type. All of the information below is based on 3.1.

Below is a list of the operators available for use in Rule Expressions. Parentheses () can be used to designate which parts of the expression should be evaluated first. $1 + 1 * 32$ equals 33. Multiplication occurs before addition. However, if you add parentheses around the addition you would have $(1 + 1) * 32$. In this example, the total is 64. Parentheses will work the same way with OpenClinica Rule Expressions.

Rule expression operators, in order of precedence:

()
* /
+ -
gt gte lt lte eq ne ct
and or

At each level of precedence the expression is evaluated from left to right.

In the examples below, we will be building the expressions with the default structure

- SED_OID = The OID depicting a particular study event definition. If the rule is referencing more than one event definition, _X, where X is a number, will be appended to the OID
- CRF_OID = The OID depicting a particular CRF. This will cover all versions of this CRF. If the rule is referencing more than one CRF, _X, where X is a number, will be appended to the OID
- CRF_VERSION_OID = The OID depicting a particular CRF Version. If the rule is referencing more than one CRF Version, _X, where X is a number, will be appended to the OID
- GROUP_OID = The OID depicting a particular Group. If the rule is referencing more than one Group, _X, where X is a number, will be appended to the OID
- ITEM_OID = The OID depicting a particular Item. If the rule is referencing more than one Item, _X, where X is a number, will be appended to the OID

Approved for publication by Jessica MacMinn. Signed on 2014-07-22 3:00PM

Not valid unless obtained from the OpenClinica document management system on the day of use.

6.7.1 Equal or Not Equal To

The following examples will demonstrate how to use both the equal (eq) and the not equal () operators with the following data types:

- INT
- REAL

- DATE
- ST
- FILE

Example 1

The first example will demonstrate how to specify a temperature is not equal to a particular negative value. The data type is INT so no decimal places are allowed.

```
<RuleDef OID="RULE1" Name="This is an example">
  <Description>The temperature is not equal to -10 degrees F</Description>
  <Expression>ITEM_OID ne (-10)</Expression>
</RuleDef>
```

Please be aware of the use of parentheses around the negative value. This is how you must write your RuleExpression if you want to evaluate negative values entered as data in the CRF. If the DataType was REAL, decimal places would be allowed.

Also, if the field was a ST, quotes (") must surround the string value. If these are not present, the RuleExpression will not evaluate correctly.

Example 2

The second example is similar, but it is stating that a value for one item is equal to the value for one item, or not equal to a value from a third item. The second item is part of the same CRF Version as the first item, but the third item is in a different event CRF. If you compare two fields together, they must be of the same DataType. If the third item is comparing an explicit value, it does not have to be the same DataType

```
<RuleDef OID="RULE2" Name="This is an example">
  <Description>Temperature is different and the color of the sun is yellow</Description>
  <Expression>SED_OID.CRF_OID.GROUP_OID.ITEM_OID_3 eq "yellow" or
CRF_VERSION_OID.GROUP_OID.ITEM_OID_2 ne ITEM_OID_1</Expression>
</RuleDef>
```

As you can see, the path will continue to grow to the left if you have to include the CRF or CRF Version, or the Study Event Definition. Each piece of the path is separated with a period (.) and everything must be capitalized.

Example 3

The following example uses DATES with the equal and not equal operators. The same principle applies where the system can only compare items of the same DataType.

```
<RuleDef OID="RULE3" Name="This is an example">
  <Description>Visit Date is the same as today's date</Description>
  <Expression>ITEM_OID_1 eq _CURRENT_DATE</Expression>
</RuleDef>
```

_CURRENT_DATE is a system level property that will pull the "today's date" from the server OpenClinica is installed on.

```
<RuleDef OID="RULE4" Name="This is an example">
```

```
<Description>The date is not December 31, 2012r</Description>
<Expression>ITEM_OID_1 ne 2012-12-31</Expression>
</RuleDef>
```

Notice the date must be in the [ISO 8601 format](#) otherwise the system will reject the RuleExpression.

Example 4

When an item is defined as a FILE DataType, you can write rules to make sure the field is either blank, or not blank. You will not be able to write any rules comparing files or saying files are equal to each other. An example of a correct RuleExpression would be:

```
<RuleDef OID="RULE5" Name="This is an example">
  <Description>no file is provided</Description>
  <Expression>ITEM_OID_1 eq ""</Expression>
</RuleDef>
```

"" next to each other in a RuleExpression signifies a blank field for all DataTypes.

As we go through the rest of the operators, we will also demonstrate how to combine multiple operators into a RuleExpression which should provide even greater clarity on how to use Rules.

6.7.2 Greater Than or Equal To

The following examples will demonstrate how to use both the greater than (gt) and the greater than or equal to (gte) operators with the following data types:

- INT
- REAL
- DATE

The 'gt' and 'gte' operators can not be used with ST or FILE.

Example 1

The first example will demonstrate how to specify a temperature is greater than a particular positive value. The data type is INT so no decimal places are allowed.

```
<RuleDef OID="RULE1" Name="This is an example">
  <Description>The temperature for this subject was greater than 105 degrees</Description>
  <Expression>ITEM_OID gt 105</Expression>
</RuleDef>
```

If the DataType was REAL, decimal places would be allowed.

Example 2

The second example is similar, but it is stating that a value for one item is greater than or equal to the value for another item. The second item is part of the same Group as the first item. Both items must be of the same DataType in order for this comparison to work. If one item is an INT and the other is DATE, the RuleExpression will be invalid.

```
<RuleDef OID="RULE2" Name="This is an example">
  <Description>Temperature is different</Description>
  <Expression>ITEM_OID_1 gte ITEM_OID_2</Expression>
</RuleDef>
```

If the second item was not part of the same group, but contained in the same CRF, the Expression would have to be written as follows:

```
<RuleDef OID="RULE3" Name="This is an example">
  <Description>Temperature is different</Description>
  <Expression>ITEM_OID_1 gte GROUP_OID.ITEM_OID_2</Expression>
</RuleDef>
```

The path will continue to grow to the left if you have to include the CRF or CRF Version, or the Study Event Definition. Each piece of the path is separated with a period (.) and everything must be capitalized.

Example 3

The following example uses DATEs with the greater than or the greater than and equal to operators. The same principle applies where the system can only compare items of the same DataType.

```
<RuleDef OID="RULE4" Name="This is an example">
  <Description>Visit Date is greater than today's date</Description>
  <Expression>ITEM_OID_1 gt _CURRENT_DATE</Expression>
</RuleDef>
```

_CURRENT_DATE is a system level property that will pull the "today's date" from the server OpenClinica is installed on.

```
<RuleDef OID="RULE5" Name="This is an example">
  <Description>The date is 2013 or later</Description>
  <Expression>ITEM_OID_1 gt 2012-12-31</Expression>
</RuleDef>
```

Notice the date must be in the [ISO 8601 format](#) otherwise the system will reject the RuleExpression.

As we go through the rest of the operators, we will also demonstrate how to combine multiple operators into a RuleExpression which should provide even greater clarity on how to use Rules.

6.7.3 Less Than or Equal To

The following examples will demonstrate how to use both the less than (lt) and the less than or equal to (lte) operators with the following data types:

- INT
- REAL
- DATE

The 'lt' and 'lte' operators can not be used with ST or FILE.

Example 1

The first example will demonstrate how to specify a temperature is less than a particular positive value. The data type is INT so no decimal places are allowed.

```
<RuleDef OID="RULE1" Name="This is an example">
  <Description>The temperature for this subject was less than 105 degrees</Description>
  <Expression>ITEM_OID lt 105</Expression>
</RuleDef>
```

If the DataType was REAL, decimal places would be allowed.

Example 2

The second example is similar, but it is stating that a value for one item is less than or equal to the value for another item. The second item is part of the same Group as the first item. Both items must be of the same DataType in order for this comparison to work. If one item is an INT and the other is DATE, the RuleExpression will be invalid.

```
<RuleDef OID="RULE2" Name="This is an example">
  <Description>Temperature is different</Description>
  <Expression>ITEM_OID_1 lte ITEM_OID_2</Expression>
</RuleDef>
```

If the second item was not part of the same group, but contained in the same CRF, the Expression would have to be written as follows:

```
<RuleDef OID="RULE3" Name="This is an example">
  <Description>Temperature is different</Description>
  <Expression>ITEM_OID_1 lte GROUP_OID.ITEM_OID_2</Expression>
</RuleDef>
```

The path will continue to grow to the left if you have to include the CRF or CRF Version, or the Study Event Definition. Each piece of the path is separated with a period (.) and everything must be capitalized.

Example 3

The following example uses DATES with the less than or the less than and equal to operators. The same principle applies where the system can only compare items of the same DataType.

```
<RuleDef OID="RULE4" Name="This is an example">
  <Description>Visit Date is less than today's date</Description>
  <Expression>ITEM_OID_1 lt _CURRENT_DATE</Expression>
</RuleDef>
```

_CURRENT_DATE is a system level property that will pull the "today's date" from the server OpenClinica is installed on.

```
<RuleDef OID="RULE5" Name="This is an example">
  <Description>The date is 2013 or earlier</Description>
  <Expression>ITEM_OID_1 lt 2012-12-31</Expression>
</RuleDef>
```

Notice the date must be in the [ISO 8601 format](#) otherwise the system will reject the RuleExpression.

As we go through the rest of the operators, we will also demonstrate how to combine multiple operators into a RuleExpression which should provide even greater clarity on how to use Rules.

6.7.4 Arithmetic Operators

OpenClinica Rules support the following arithmetic operators:

- + (Addition)
- - (Subtraction)
- * (Multiplication)
- / (Division)

All four of the operators above will work with INT and REAL DataTypes. For DATES, an INT can be added (+) or subtracted (-) from the DATE to increase or decrease it by a certain number of days. Subtraction (-) can also be used to get the difference between two DATES, with the result being an INT representing the number of days between the two DATES. The difference between dates is always computed as an absolute number. For example, 2011-11-19 - 2011-11-20 evaluates to 1, not -1. ST fields can not be used with any of the arithmetic operators.

Example 1

```
<RuleDef OID="RULE1" Name="This is an example">
  <Description>The average temperature is greater than 98.6</Description>
  <Expression>((ITEM_OID_1 + ITEM_OID_2) / 2) gt 98.6</Expression>
</RuleDef>
```

The above example is taking 2 fields that are REAL DataTypes, ITEM_OID_1 and ITEM_OID_2, adding them together to get a total, and then dividing by 2. If the average is greater than 98.6, this Rule will fire.

Example 2

Addition and subtraction with dates can be very powerful, but there are a few things to keep in mind. When working with dates and these arithmetic operators, the values derived will be in days. For example, you can not have a rule that checks for a visit that is greater than 1 year. You would have to write this rule saying the difference between the dates is greater than 365 days.

```
<RuleDef OID="RULE2" Name="This is an example">
  <Description>The surgery date is greater than 1 year from this visit date</Description>
  <Expression>ITEM_OID_1 - ITEM_OID_2 gte 365</Expression>
</RuleDef>
```

In the above RuleExpression, we are subtracting one date from another, and if the value is 365 or greater, the Rule should evaluate to true.

Example 3

OpenClinica computes addition/subtraction expressions from left to right. When combining dates and integers in an expression this has some interesting consequences. A date that has an integer added to or subtracted from it is still treated as a date. When two dates are compared, it uses the absolute value of the difference (ie, the result is always a positive number or 0). Thus the two expressions below are not always equivalent:

A) $0 - 1 + \text{DATE2} - \text{DATE1} \neq 0$

B) $0 + \text{DATE2} - \text{DATE1} - 1 \neq 0$

Assume $\text{DATE2} = 31\text{-Dec-2000}$ and $\text{DATE1} = 01\text{-Jan-2001}$

In case A, -1 is added to DATE2 and the result is a date that is one day before DATE2. OpenClinica then takes that date and subtracts DATE1 from it, and since for date math we use absolute values, computes the absolute value of the difference between the two dates.

A) $0 - 1 + \text{DATE2} - \text{DATE1} \neq 0$

$(-1) + 31\text{-Dec-2000} - \text{DATE1} \neq 0$

$30\text{-Dec-2000} - 01\text{-Jan-2001} \neq 0$

$2 \neq 0$

In case B, OpenClinica first computes the absolute value of the difference between DATE2 and DATE1, then subtracts 1.

B) $0 + \text{DATE2} - \text{DATE1} - 1 \neq 0$

$31\text{-Dec-2000} - 01\text{-Jan-2001} - 1 \neq 0$

$1 - 1 \neq 0$

$0 \neq 0$

6.7.5 6.7.5 The Contains Operator

The following examples will demonstrate how to use the contains (ct) operator with the following data types:

- ST
- INT
- REAL
- DATE
- FILE

Example 1:

The first example will demonstrate how, when using checkbox or multi-select items with a data type of ST, to check if a certain option is selected. For example, if you have a checkbox item with three options (Option 1, Option 2, and Unknown), you may want to make sure that 'Option 1' and 'Option 2' are not selected if 'Unknown' is selected:

```
<RuleDef OID="RULE1" Name="This is an example using ST data type">
  <Description>'Unknown' and at least one other option selected</Description>
  <Expression>(ITEM_OID_1 ct "1" or ITEM_OID_1 ct "2") and ITEM_OID_1 ct
"99"</Expression>
</RuleDef>
```

Similarly, the contains operator can be used with data types INT and REAL. Please note that all values must be enclosed in quotation marks in the expression when using the ct operator (i.e. "1" and "2").

Example 2:

This example will demonstrate how to use the contains operator to check for a certain year when using an item of data type DATE. For example you may want to write a Rule that will verify that the date of informed consent occurred during the year 2014:

```
<RuleDef OID="RULE2" Name="This is an example using DATE data type">  
  <Description>The year is 2014</Description>  
  <Expression>ITEM_OID_1 ct "2014"</Expression>  
</RuleDef>
```

When set to evaluate to true, the Rule will fire if the date occurs in the year 2014. Alternatively, you could set the Rule to evaluate to false so that the Rule fires if the date does not occur in the year 2014.

Example 3:

When an item is defined as a FILE data type, a Rule can be written to make sure the filename is a certain extension type. For example, the contains operator can be used to check for a PDF file:

```
<RuleDef OID="RULE3" Name="This is an example using FILE data type">  
  <Description>The extension must be .pdf</Description>  
  <Expression>ITEM_OID_1 ct ".pdf"</Expression>  
</RuleDef>
```

If written to evaluate to false, the Rule will fire and an error message will appear if a file type other than .PDF is uploaded.