

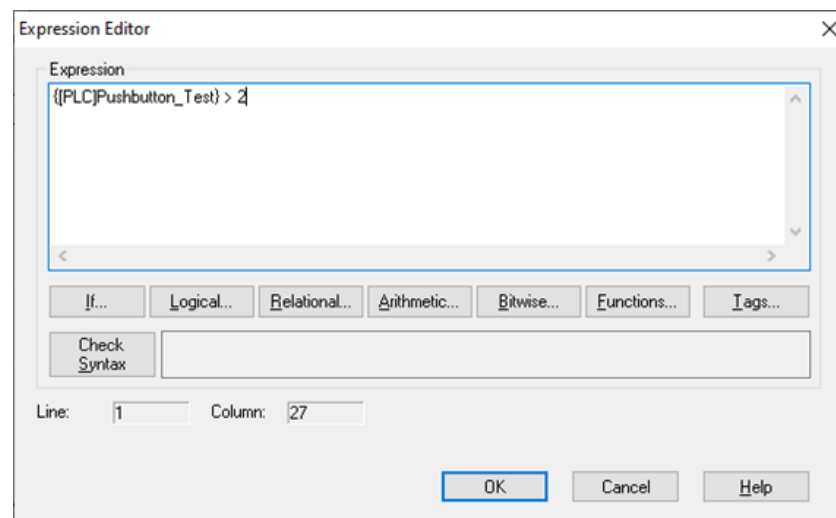
Using Expressions

Dan Kandray

January 30, 2020

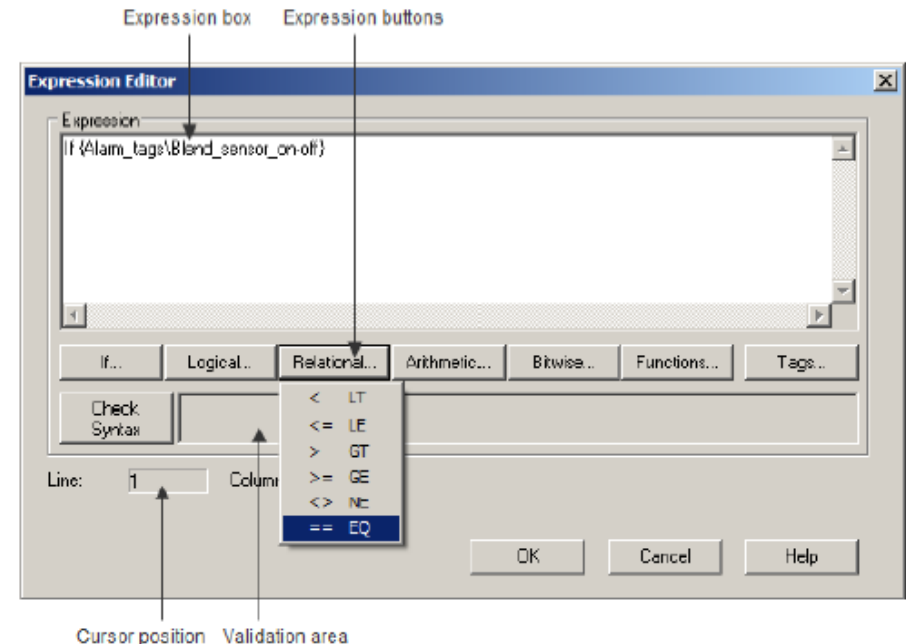
Using Expressions

- Many types of animation can be achieved using expressions.
- You can use expressions containing tag values, constants, mathematical equations, security functions, and if-then-else logic.
- A tag name or tag placeholder can be included as part of an expression, or it can stand alone as the entire expression.
- The Expression Editor is used to create expressions.



Expressions

- An expression can be a tag name, a tag placeholder, a mathematical or logical equation, or a function that returns a numeric, string, or true or false value.
 - For expressions that return true and false values, 1 and other non-zero values signify True, and zero signifies False.
- Expressions let you manipulate the data gathered from devices and make it more meaningful by comparing it to other values, combining it with other values, creating cause and effect relationships between values, or displaying it.



Expressions can be used in these editors: *Swagelok*

- Graphics — You can define an expression to control various aspects of a graphic object's appearance.
- Alarm Setup — When setting up alarms, you can use expressions for alarm triggers, and with some of the connections that silence and acknowledge alarms.
- Information Setup — You can use expressions to determine when to show information messages.

Expressions can contain:

- Tags, tag placeholders, arithmetic, and bitwise operators, and mathematical functions such as SQRT (square root), that return numeric values.
- Relational and logical operators, and functions such as ALM_FAULT (alarm fault), that return true/false values.
- If-Then-Else logic that returns numeric or true and false values, depending on how it's structured.
 - These are called conditional expressions because the result of the expression depends on whether the If statement is true or false.
 - When the If statement evaluates to True the result is defined by the Then statement. When the If statement is False the result is defined by the Else statement.

```
if (tag1 > tag2) then 0 else if (tag1 > tag3) then 2 else 4
```

- A tag name can be included as part of an expression or can stand alone as the entire expression.
- A constant can have any of the following formats:
 - Integer (123)
 - Floating-point (123.45)
 - String constant ("character string")
- Arithmetic operators perform math on two or more numeric values and calculate the result.
 - The arithmetic operators include: +, -, *, /, MOD (remainder), ** (exponent)

Expression Formats

- A tag name can be included as part of an expression or can stand alone as the entire expression.
- A constant can have any of the following formats:
 - Integer (123)
 - Floating-point (123.45)
 - String constant ("character string")
- Arithmetic operators perform math on two or more numeric values and calculate the result.
 - The arithmetic operators include: +, -, *, /, MOD (remainder), ** (exponent)
- String operands:
 - The + operator can be used to join string operands. For example, the expression "hello" + "world" returns: helloworld.
 - You cannot join string tags to analog tags, whether they are HMI or data server tags.

Relational operators

- Relational operators compare two numeric or string values to provide a true or false result. If the statement is true, a value of 1 is returned. If false, 0 is returned.

Symbols	Operator	Numeric Example	String Example
For the numeric examples, tag1 = 5 and tag2 = 7 For the string examples, serial_no = "ST009"			
EQ, ==	equal	tag1 == tag2 false	serial_no == "ST009" true
NE, <>	not equal	tag1 <> tag2 true	serial_no <> "ST011" true
LT, <	less than	tag1 < tag2 true	serial_no < "ST011" true
GT, >	greater than	tag1 > tag2 false	serial_no > "ST011" false
LE, <=	less than or equal to	tag1 <= tag2 true	serial_no <= "ST011" true
GE, >=	greater than or equal to	tag1 >= tag2 false	serial_no >= "ST011" false

Logical operators

- Logical operators determine the validity of one or more statements.
 - AND, OR, and NOT.
 - The operators return a non-zero value if the expression is true, or a 0 if the expression is false.
- Any statement that evaluates to a non-zero value is regarded as true.
 - For example, the statement tag1 is false if the value of tag1 is 0, and true if tag1 has any other value.

Symbols	Operator	Action	Example (For these examples, tag1 = 5 and tag2 = 7)
AND, &&	and	Returns a 1 if the statements to the right and left of the operator are both true.	(tag1 < tag2) AND (tag1 == 5) Both statements are true; returns a 1.
OR,	or	Returns a 1 if either the statement to the left or right of the operator is true.	(tag1 > tag2) OR (tag1 == 5) tag1 == 5 is true; returns a 1.
NOT	negation	Reverses the logical value of the statement it operates on.	NOT (tag1 < tag2) Although tag1 < tag2 is true, NOT reverses the logical value; returns a 0.

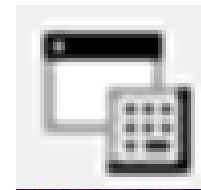
Mathematical functions

This function	Returns this value
SQRT (expression)	The square root of the expression
LOG (expression)	The natural log of the expression
LOG10 (expression)	The base ten log of the expression
SIN (expression)	The sine of the expression in radians
COS (expression)	The cosine of the expression in radians
TAN (expression)	The tangent of the expression in radians
ARCSIN (expression)	The arc sine of the expression in radians
ARCCOS (expression)	The arc cosine of the expression in radians
ARCTAN (expression)	The arc tangent of the expression in radians
SIND (expression)	The sine of the expression in degrees
COSD (expression)	The cosine of the expression in degrees
TAND (expression)	The tangent of the expression in degrees
ARCSIND (expression)	The arc sine of the expression in degrees
ARCCOSD (expression)	The arc cosine of the expression in degrees
ARCTAND (expression)	The arc tangent of the expression in degrees

- If-then-else expressions carry out an action conditionally or branch actions depending on the statements in the expression.
- The if-then-else statements enable the expression to perform different actions in different situations and to repeat activities until a condition changes.
- To build conditional expressions, use the relational operators and the logical operators for the statement and values.
- The if-then-else structure is:
 - if statement then value1 else value2

Use write expressions

- Write expressions allow the operator to enter a value which is manipulated by an expression before being sent to the data source.
- FactoryTalk View substitutes the value the operator enters for the placeholder in the expression, calculates the value of the expression, and writes the result to the Value connection.
- All write expressions must contain a question mark (?) as a placeholder for the value the operator enters.
- Use with the Numeric Input Enable button and the numeric input cursor point. When the operator presses the button or cursor point, a keypad or scratchpad opens.



Example: Use write expressions

- In this example, the operator regulates the speed of a conveyor belt by entering a value in feet or meters per second. When the operator enters the value in meters per second, the value is ***converted to feet per second*** before being passed to the data source.
- The operator first indicates whether the value is in feet or meters by pushing a Maintained push button. The push button has one state corresponding to feet per second, and the other state to meters per second.
- **First set up the Maintained push button :**
 1. In the **Maintained Push Button Properties** dialog box, on the **States** tab, set up these states:
 - **State 0** — Value: 0, Caption: Feet/min
 - **State 1** — Value: 1, Caption: Meters/min
 2. On the **Connections** tab, assign a digital tag called Feet_or_meters to the **Value** connection (either an HMI tag or a data server tag).

Example: Use write expressions

Second, set up the Numeric Input Enable button:

1. On the **Label** tab of the **Numeric Input Enable Properties** dialog box, type the caption *Enter conveyor speed*.
2. On the **Connections** tab, assign a tag called *Box_Display_Timer.PRE* to the **Value** connection.

3. Assign this expression to the **Optional Exp** connection:

If {[PLC]Feet_OR_Meters} == 0 Then (12/?*1000*60) Else (3.66/?*1000*60)

- FactoryTalk View writes the result of the expression to the Conveyor_speed tag at the data source.