

Operators

Operators in expressions

The following tables show operand types and results of arithmetical and logical operations.

- [arithmetical operators](#)
- [logical operators](#)
- [relational operators](#)

When evaluating any expressions, there holds that if any operand in a calculation is invalid, the result is also invalid. However, an eval tag is exception, if the parameter "Replace Invalid values with 0" is set. Then it is evaluated as it is mentioned [here](#).

See also the topic [Order of operators by the priority](#)

Arithmetical operators

Operator	Operation	Description	Operand types	Result type	
+	addition	Difference between two absolute times is calculated this way: ATime - BTime = %SubTimeLocal(ATime, BTime) All others combination of absolute and relative times with operators + and - are calculated this way: ATime + RTime = %AddTimeLocal(ATime, RTime) or ATime - RTime = %AddTimeLocal(ATime, -RTime)	integer real boolean time interval	integer real boolean time interval	
			Operand 1	Operand 2	absolute time
			absolute time	integer real relative time	
			integer real relative time	absolute time	
-	subtraction	Difference between two absolute times is calculated this way: ATime - BTime = %SubTimeLocal(ATime, BTime) All others combination of absolute and relative times with operators + and - are calculated this way: ATime + RTime = %AddTimeLocal(ATime, RTime) or ATime - RTime = %AddTimeLocal(ATime, -RTime)	integer real boolean time interval	integer real boolean time interval	
			Operand 1	Operand 2	absolute time
			absolute time	integer real relative time	
*	multiplication		integer real boolean	integer real integer	
/	division		integer real boolean	real real integer	
-	unary operator		integer real	integer real	

In mathematical operations, operands of Boolean type are converted to Integer type as follows:

- TRUE -> 1
- FALSE -> 0

Note

- Result of an function of Boolean type may be converted to other type by the multiplying with the constant of the particular type. For example: %Flag(Object, @A)*1 gives a numerical type of the result value, i.e. Integer, Real.

Logical operators

operator	operation	operand types	result type
&	logical multiplication	boolean integer	boolean integer
	logical addition	boolean	boolean
!	negation	boolean	boolean

Logical multiplication of two operands of **INTEGER** type represents decade form of logical multiplication performed between the corresponding bits of the both operands binary forms.

Example

179 & 217 = 145

1	0	1	1	0	0	1	1		179
1	1	0	1	1	0	0	1		217
1	0	0	1	0	0	0	1		145

Relational operators

operator	operation	operand types	result type
=	equal	integer real absolute time time interval	boolean
#	unequal	integer real absolute time time interval	boolean
>	greater	integer real absolute time time interval	boolean
<	less than	integer real absolute time time interval	boolean
>=	greater or equal	integer real absolute time time interval	boolean
<=	less or equal	integer real absolute time time interval	boolean