Expressions

Operators , Order of Precedence, Predefined Constants, Predefined Time Constants, Predefined Reset Options, Math Functions, Logical Functions, String Functions, Conversion Functions, Time Functions, Statistical Functions

Components that display data values either numerically or graphically can be processed using expressions. These expressions can include simple mathematical expressions, functions to manipulate strings, or more complex functions that deal with the state of a data value over time.

For instance, a temperature reading in degrees Celsius can be processed to display in degrees Fahrenheit using a simple mathematical expression. This is done by first selecting the data value in the **Select Data** field, and then entering the mathematical expression after the defined data value. Using the above example, if the data value is defined as "Server:CR5000.TempData.Temp1" ("*Source*:*datalogger.table.variable"*), you would enter

"Server:CR5000.TempData.Temp1" * 1.8 + 32

to convert the temperature reading from degrees Celsius to degrees Fahrenheit.

Strings

As shown above, double quotes are used in Web Publisher to enclose the name of a data value (or source, datalogger, or table depending on the component). Therefore, when defining a literal string, a dollar sign is used as a prefix. This indicates to Web Publisher that you are defining a literal string rather than a data value. For example, to search for the position of the sequence abc in the data value mystring, you would use the following expression:

InStr(1, "Server:CR1000.hourly.mystring", $"abc")

Statistical Functions

Expressions can also use Statistical Functions, some of which involve the state of a data value over a period of time. For instance, you can return the maximum value of a data value over the past 24 hours using the expression:

MaxRunOverTime("Server:CR1000.QtrHour.Temp",Timestamp ("Server:CR1000.QtrHour.Temp"),nsecPerDay)

Aliases

If a data value is used multiple times in an expression, the expression can be simplified by declaring an alias for the data value at the first of the expression, in the form:

Alias(alias_name, data_value)

For example,

IIF(ABS(("Server:CR1000.MyTable.Value"-ValueAtTime("Server:CR1000.MyTable.Value",TimeStamp ("Server:CR1000.MyTable.Value"),30*nsecPerSec,0))>10 AND ABS(ValueAtTime ("Server:CR1000.MyTable.Value",TimeStamp("Server:CR1000.MyTable.Value"),30*nsecPerSec,0)-ValueAtTime("Server:CR1000.MyTable.Value",TimeStamp ("Server:CR1000.MyTable.Value"),60*nsecPerSec,0)))>10,1,0)

can be replaced by:

Alias(X,"Server:CR1000.MyTable.Value");IIF((ABS(X-ValueAtTime(X,TimeStamp(X),30*nsecPerSec,0))>10 AND ABS(ValueAtTime(X,TimeStamp(X),30*nsecPerSec,0)-ValueAtTime(X,TimeStamp(X),60*nsecPerSec,0)))>10,1,0)

All of the functions available in Web Publisher are described below. For details on a function, click on the function name.

Notes: Spaces must be used to delimit the predefined constants and functions. Operators allow but do not require spaces.

An expression can include data values from multiple dataloggers.

Operators

	Description
( )	Prioritizes an expression
*	Multiply by
/	Divide by
^	Raised to the power of
+	Add
-	Subtract/Unary negation
=	Equal
<>	Not equal
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

Order of Precedence

- Anything inside parentheses ( )
- Exponentiation ^
- Negation (unary) -
- Multiplication *, division /
- Modulo (remainder) MOD
- Addition +, subtraction -

When consecutive operators have the same priority, the expression evaluates from left to right. This means that an expression such as **a-b-c** is evaluated as **(a-b)-c**.

Predefined Constants

The following constants are defined for convenience within numeric expressions:

Constant	Description
e	2.718282
PI	3.141593
True	-1
False	0
NOPLOT	NAN
NAN	NAN (not a number)
INF	INF (non-finite number)

Predefined Time Constants

These predefined time constants can be useful as a parameter for the Statistical Functions, where the interval parameter must be specified in nanoseconds.

Constant	Description
nsecPerUSec	Number of nanoseconds in a microsecond
nsecPerMSec	Number of nanoseconds in a millisecond
nsecPerSec	Number of nanoseconds in a second
nsecPerMin	Number of nanoseconds in a minute
nsecPerHour	Number of nanoseconds in an hour
nsecPerDay	Number of nanoseconds in a day
nsecPerWeek	Number of nanoseconds in a week

Predefined Reset Options

These predefined reset options are used as a parameter for the Statistical Functions with a reset parameter.

Constant	Description
RESET_HOURLY	Reset whenever there is a change in the hour in the value's timestamp
RESET_DAILY	Reset whenever there is a change in the day in the value's timestamp
RESET_WEEKLY	Reset whenever the day of the week is less than the day of the week of the newest timestamp stored (Sunday marks the beginning of the week) or when the difference between the current timestamp and the newest timestamp stored exceeds seven days.
RESET_YEARLY	Reset whenever there is a change in the year in the value's timestamp
RESET_CUSTOM	Reset whenever the doReset parameter is set to a non-zero value.

Math Functions

Function	Description
ABS	Returns the absolute value of a number.
ACOS	Returns the arc cosine of a number.
ASIN	Returns the arc sine of a number.
ATN	Returns the arc tangent of a number.
ATN2(y,x)	Returns the arctangent of y/x.
CEILING	Rounds a number up to an integer value.
COS	Returns the cosine of a number.
COSH	Returns the hyperbolic cosine of a number.
CSGN	Changes the sign of a number by multiplying by -1.0.
EXP	Returns e raised to a power.
FIX	Returns the integer portion of a number. If the number is a negative, the first negative integer greater than or equal to the number is returned.
FLOOR	Rounds a number down to an integer value.
FRAC	Returns the fraction part of a number.
FormatFloat	Converts a floating point value into a string.
FormatFloatL	Converts a floating point value into a string and applies any rules associated with the locale of the computer running Web Publisher.
INT	Returns the integer portion of a number. If the number is a negative, the first negative integer less than or equal to the number is returned.
IsFinite	Determines if a value is finite.
LN	Returns the natural log of a number. (Note that LN or LOG may be used to perform the same function.)
LOG	Returns the natural log of a number. (Note that LN or LOG may be used to perform the same function.)
LOG10	Returns the logarithm base 10 of a number.
MOD	Performs a modulo divide of two numbers.
PWR	Raises a constant to a specified exponent.
RND	Generates a random number.
ROUND	Rounds a number to a higher or lower number.
SGN	Used to find the sign value of a number (-1, 0, or 1).
SIN	Returns the sine of an angle.
SINH	Returns the hyperbolic sine of a number.
SQR	Returns the square root of a number.
TAN	Returns the tangent of an angle.
TANH	Returns the hyperbolic tangent of a number.

Logical Functions

Function	Description
AND	Performs a logical conjunction on two numbers.
EQV	Performs a logical equivalence on two numbers.
IIF(x,y,z)	Evaluates an expression and returns one value if true, a different value if false.
IMP	Performs a logical implication on two numbers.
NOT	Performs a logical negation on a number.
OR	Performs a logical disjunction on two numbers.
SelectSwitch	Iterates through the set of predicates and values in the order in which these are specified in its arguments list. It will return the value associated with the first predicate that specifies a non-zero integer value. If no asserting predicate can be found, the function will return the default_value
XOR	Performs a logical exclusion on two numbers.

String Functions

Function	Description
Hex	Returns a hexadecimal string representation of an expression.
HexToDec	Converts a hexadecimal string to a float or integer.
InStr	Finds the location of a string within a string.
InStrRev	Finds the location of a string within a string. (Differs from InStr in that it searches from the end of the string rather than from the start of the string.)
Left	Returns a substring that is a defined number of characters from the left side of the original string.
Len	Returns the number of bytes in a string.
LTrim	Returns a copy of a string with no leading spaces.
Mid	Returns a substring that is within a string.
Replace	Used to search a string for a substring, and replace that substring with a different string.
Right	Returns a substring that is a defined number of characters from the right side of the original string.
RTrim	Returns a copy of a string with no trailing spaces.
Space	Returns a string value that is filled with a defined number of spaces
StrComp	Compares two strings by comparing the characters in one string to the characters in another.
StrReverse	Returns a copy of a string with the characters in reverse order.
Trim	Returns a copy of a string with no leading or trailing spaces.

Conversion Functions

Function	Description
ToDate	Converts a value to a date.
ToFloat	Converts a value to a floating point number.
ToInt	Converts a value to an integer.

Time Functions

Function	Description
FormatTime	Produces a string that formats a timestamp in the manner specified.
SystemTime	Returns the current computer time.
SystemTimeGMT	Returns the current GMT (Greenwich Mean Time) system time.
Timestamp	Returns the timestamp associated with the record from which a value is derived.
SetTimestamp	Returns the value specified and sets its timestamp to the timestamp specified.

Statistical Functions

Function	Description

AvgRun	Returns a running average of up to the last specified number of values.
AvgRunOverTime	Returns the running average of the specified value over time.
AvgRunOverTimeWithReset	Returns the running average of the specified value since the function was reset.
AvgSpa	Returns the average of the specified values.
Last	Stores the specified value and returns the previous value.
MaxRun	Returns the maximum of all values that it has considered.
MaxRunOverTime	Returns the maximum of all values whose timestamps are greater than the newest timestamp minus the specified interval.
MaxRunOverTimeWithReset	Returns the maximum of all values since the function was reset.
MaxSpa	Returns the maximum of the specified values.
MedianRun	Returns the median value of up to the last specified number of values.
MedianRunOverTime	Returns the median value in the set of values whose timestamps are greater than the newest timestamp minus the specified interval.
MinRun	Returns the minimum of all values that it has considered.
MinRunOverTime	Returns the minimum of all values whose timestamps are greater than the newest timestamp minus the specified interval.
MinRunOverTimeWithReset	Returns the minimum of all values since the function was reset.
MinSpa	Returns the minimum of the specified values.
StdDev	Returns the standard deviation of up to the last specified number of values.
StdDevOverTime	Returns the standard deviation of the specified value over time.
StdDevOverTimeWithReset	Returns the standard deviation of the specified value since the function was reset.
Total	Returns the total of all values that it has considered.
TotalOverTime	Returns the total of all values whose timestamps are greater than the newest timestamp minus the specified interval.
TotalOverTimeWithReset	Returns the total of all values since the function was reset.
ValueAtTime	Returns the oldest value in a set of values from a specified time interval.