

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

### Logical Functions

Function	Description
<a href="#">AND</a>	Performs a logical conjunction on two numbers.
<a href="#">EQV</a>	Performs a logical equivalence on two numbers.
<a href="#">IIF(x,y,z)</a>	Evaluates an expression and returns one value if true, a different value if false.
<a href="#">IMP</a>	Performs a logical implication on two numbers.
<a href="#">NOT</a>	Performs a logical negation on a number.
<a href="#">OR</a>	Performs a logical disjunction on two numbers.
<a href="#">SelectSwitch</a>	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
<a href="#">XOR</a>	Performs a logical exclusion on two numbers.

### String Functions

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

### Conversion Functions

Function	Description
<a href="#">ToDate</a>	Converts a value to a date.
<a href="#">ToFloat</a>	Converts a value to a floating point number.
<a href="#">ToInt</a>	Converts a value to an integer.

### Time Functions

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

### Statistical Functions

Function	Description
----------	-------------

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