

Calculated Variables using Formulas' and @Functions

A Calculated variable has its value derived based on a formula that you specify.

C-calculated variables do not appear on the Add/Edit/Insert/Copy screens as the value is determined when the Save key is pressed.

Calculated variables can be charted, used in other calculations, and can have rules and limits assigned to it.

The length of the variable must include an allowance for a decimal point and sign if required.

Calculation:

Click on the Formula button to display a popup of available @functions.

Name: ... Type:

Functions	Description
@ABS[formula]	Absolute value of formula
@ATAN[formula]	Calc. ArcTan of formula (Radians)
@AVG[V##..V##]	Return AVERAGE of variable range
@CASE[st,mn,mx,val,...]	Return value if stmt within min/max range
@CHOOSE[V##,list]	Return LIST item based on V## value (0..99)
@COMPLYCL[V##..V##]	Return % meeting compliance to Control Limits
@COMPLYSL[V##..V##]	Return % meeting compliance to Spec. Limits
@COS[formula]	Calc. COS of formula (Radians)
@COUNT[V##..V##]	COUNT non-null entries in variable range
@DAY[]	Return DAY based on current record
@DOW[]	Return DAY OF WEEK based on current record (1=

- V1 - Date
- V2 - Time
- V3 - Auto_Shift_Id
- V4 - Shift

The Right mouse click to display a list of all the variables you can use in a formula

QWAdmin will scan the formula, when the Save button is clicked, to ensure the formula has balanced sets of brackets [] and parenthesis ()

If @functions are used in a formula, they are checked against a list of valid functions.

If an error is detected in the formula you will not be able to save the variable until the formula is corrected.

Creating a Calculated Variable using a Formula

A formula can be as simple as a constant such as "CALC: 14", which sets the variable to a value of "14".

Calculated Variables using Formulas' and @Functions

A formula can also be a complex combination of constants, mathematical operators (+ - / * ^), V##- variable reference numbers, QW @ functions and parenthesis ().

Calculated variables make use of Input and Output files. In the following case V17, a Calculated variable, is used as a counter.

INPUT FILENAME... SHIFTTOT.V17

OUTPUT FILENAME.. SHIFTTOT.V17

CALC: V17 + V8

At the time the Add screen is selected, QW reads the value stored in SHIFTTOT.V17, example 128, and assigns V17 the value 128. When the Insert key is pressed to save the entered data the value of V8, example 13, is added to the value of V17, 128. The new value of $128 + 13 = 141$ is stored in the OUTPUT FILENAME SHIFTTOT.V17. The next time the Add data key is pressed the value of 141 is stored in V17 and the process is repeated.

Note: QW6 computes calculated variables in a set sequence, from the lowest variable number to the highest, V3, V4,V5..V99. If you are using a Calculated variable in the formula of a second Calculated variable make sure the one used in the formula has a lower V## or the result of the calculation will be incorrect.

The following is a description of the conventions you must follow to create Calculated variables and maximize their use.

Operator	Description	Example
+	adds two values together	V16 + 23
-	subtracts one value from another	V16 - 23
*	multiplies one value by another	V16 * 1.33
/	divides one value by another	V16 / V12
^	raises a value to the power of another	V12^2

V##

Variable reference numbers are used in QW to give you access to the values of other variables entered or computed in the same record or entry, or previous values in the case of special @ Moving value or Historical Record functions.

You can use variable types Numeric, Calculated, Average, Range and Popup in a formula. A Popup variable used in a formula will return the value of the key selected for that popup.

Calculated Variables using Formulas' and @Functions

Parenthesis ()

Use parenthesis to control the order in which QW calculates the formulae you input. QW follows the standard conventions on order of operations. They are multiplication, division and rising to the power will be computed in the order they occur followed by addition and subtraction in the order they occur. Depending how you use, or not use, parenthesis can render a result you may not expect.

For example:

$$3+4*6^2 = 379$$

$$((3+4)*6)^2 = 1764$$

$$3+((4*6)^2) = 579$$

$$(3+4)*(6^2) = 252$$

@ Functions

QW has a number of pre-defined functions that you can make use of to simplify the formulas you define. These functions have specific formats that you must follow. A QW function is preceded by the @ symbol to indicate it is a function. All functions have a function name followed by an expression, and in the case of Moving Functions, a restriction. The expression and restriction if any, are enclosed in square brackets [].

@FUNCTION NAME[expression, restrictions if any]

The expression can be as simple as just one variable

@SQRT[V18] - the square root of the value of V18

@AVG[V3,V4,V5,V6,V7,V8,V12] - the average value

or, it can be a complex formula including other @Functions.

@ABS[V18 - @AVG[V3,V4,V5,V6,V7,V8,V12]]

All V##'s in a list must be separated by a comma ",". In the case of the above example you can use the dot dot, "..", feature to simplify the formula.

@AVG[V3..V8, V12] - computes the same result.

The ".." feature can be used on all consecutive variables in a list.

One @ Function can be used within another @ Function eliminating the need for intermediate variables.

V16=@AVG[V4..V8]

V17=@MIN[V16,10]

These two variables can be replaced by V17=@MIN[@AVG[V4..V8],10]

Calculated Variables using Formulas' and @Functions

There are two types of @ Functions:

Historical Record @Functions (or moving value functions)

Use the values for a particular variable back in time a fixed number of entries (restriction). The number of historical entries to be used is defined in the @ Moving Function after the V##.

@MAVG[V16,5]

In this example QW will compute the average value of V16 using the last 5 records including the current one plus the four previous entries.

The minimum number of historical values in an @ Moving Function is 2.

The maximum number of historical values in an @ Moving Function is 99 or the number of records (y) currently in memory (Record x of y).

Single Record @Functions

Use the values from variables entered or calculated for a single Date and Time entry in the ADD or Log screen. Examples of these are @AVG, @RANGE and @TARGET.

QW automatically bypasses null (no value entered) values in a V## list for an @Function.

For example in the formula CALC: @AVG[V16..V18] where V16=12 V17=13 V18=null

QW will add $12 + 13 = 25$ and divide by 2 not 3 because V18 was a null value.

Available @Functions

The following is a list of QW Functions with a brief description and example of each.

Note:

Where [expression] is used it means a function allows complex calculations within the brackets, including other @ Functions.

Where [V##] is used a single V## or numeric constant is permissible.

Where [V##..V##] is used a list or range of V##'s and or numeric constants is accepted.

Where [] follows a function name no expression is required but the brackets are required.

Trigonometric Conversions:

For @ATAN, @COS, @SIN and @TAN trigonometric functions, converting to degrees or radians can be accomplished as follows:

Degrees to radians ® multiply the angle (degrees) by $\text{PI}/180$.

Radians to degrees ® multiply the angle (radians) by $180/\text{PI}$.

Calculated Variables using Formulas' and @Functions

@ Function	Description / Example
@ABS[expression]	Returns the absolute value of the expression Ex: @ABS[V5-(V14+V18)]
@ATAN[V##]	Returns the ARC TANGENT of angle V## given in radians. Ex: @ATAN[V13]
@AVG[V##..V##]	Returns the average value of the variables in the list. Ex: @AVG[V4..V6,V9,V10]
@CASE[st,mn,mx,val,...]	Uses the result of the statement and returns the assigned value that matches the min/max range Ex: @CASE[@HOUR[]+(@MIN[]/60),7.50,15.49,10,15.50,23.49,20,23.50,24.00,30,00.01,7.59,30] would return: a 10 for the 7:30 to 3:30 shift a 20 for the 3:30 to 11:30 shift a 30 for the 11:30 to 7:30 shift
@CHOOSE[V##,LIST]	Returns the entry from the LIST depending on the value of V##. Ex: @CHOOSE[V6,10,20,30] if V6=0 10 is returned if V6=1 20 is returned if V6=2 30 is returned if V6 > the number of entries in the list then null (no entry) is returned.
@COMPLYCL%[V##..V##]	Returns the percentage of variables that meet compliance to Control Limits. At least one Control Limit must be Fixed. This function is calculated as: the number of variables that have a value minus the number of variables found to be outside of Control Limits divided by the total number of variables times 100. Ex: @COMPLYCL%[V12..V18]
@COMPLYSL%[V##..V##]	Returns the percentage of variables that meet compliance to Specifications Limits. At least one Specification Limit must be Fixed.

Calculated Variables using Formulas' and @Functions

	<p>This function is calculated as: the number of variables that have a value minus the number of variables found to be outside of Specification Limits divided by the total number of variables times 100.</p> <p>Ex: @COMPLYSL%[V12..V18]</p>
@COS[V##]	<p>Returns the COSINE of angle V## given in radians.</p> <p>Ex: @COS[V13]</p>
@COUNT[V##..V##]	<p>Returns the number of non-blank,non-null variables in the V## list.</p> <p>Ex: @COUNT[V12..V18]</p>
Date Functions	<p>Returns the year of the week of the current record</p>
@YEAR[]	<p>Returns the month of the week of the current record</p>
@MONTH[]	<p>Returns the day of the week of the current record</p>
@DAY[]	<p>Returns the day of the week of the current record where:</p>
@DOW[]	<p>Sun=1 Mon=2 Tue=3 Wed=4 Thu=5 Fri=6 Sat=7</p> <p>Ex: July 23, 2002 would return YEAR=02 MONTH=07 DAY=23 and DOW=03</p>
@EXP[expression]	<p>Returns the natural anti Log value for the expression (e^x).</p> <p>Ex: @EXP[V12*3]</p>
@IF[condition,T,F]	<p>Returns either the value for True or False depending on the condition.</p> <p>Ex: @IF[V5>99,99,V5]</p> <p>If V5 is greater than 99 then the True value 99 is returned.</p> <p>If V5 is less than or equal 99 then the False value and the value of V5 is returned.</p> <p>Conditional IF Operators are:</p> <ul style="list-style-type: none">> greater than>= greater than or equal to< less than<= less than or equal to= equal to <p>A Practical @IF example to accumulate downtime minutes by shift where:</p> <p>V3 = Shift code</p>

Calculated Variables using Formulas' and @Functions

	V9 = Minutes Down
	V11 = Accumulated Minutes Down
	The calculation would be @IF [V3 = @MPREV[V3,5], V9 + @MPREV[V11,5],V9]
@INT[expression]	Returns the integer or whole number Ex: @INT[(V4*1.23)/4]
@ISNA[V##]	Tests a variable for the value null. If the value is null it returns a value 1. If the value is not null the value 0 is returned. Ex: @IF[@ISNA[V5],10,V5] If V5 is equal to null @ISNA returns a value of 1 which to the @IF function is the value for True so 10 is returned.
@LN[expression]	Returns the natural log for the expression. Ex: @LN[V10/2]
@LCL[V##]	Returns the Lower Control Limit value for a variable. Ex: @LCL[V16]
@LOG[expression]	Returns the LOG base 10 value for the expression. Ex: @LOG[V10/2]
@LSL[V##]	Returns the Lower Specification Limit value for a variable. Ex: @LSL[V16]
@MAVG[V##..V##,nn]	Returns the Moving Average for variable V## back nn historical points. The minimum value for historical points is 2. Ex: @MAVG[V9..V11,5]
@MAX[V##..V##]	Returns the maximum or highest value from the variables in the list. Ex: @MAX[V15,V18,V21]
@MC%OOL[V##..V##,nn]	Returns the percentage of values calculated to be outside the control limits for variable V## back nn historical points. The minimum value for historical points is 2. Ex: @MC%OOL[V9..V11,5]
@MCOUNT[V##..V##,nn]	Returns the number of non-null values for variable V## back nn historical points.

Calculated Variables using Formulas' and @Functions

	<p>The minimum value for historical points is 2.</p> <p>Ex: @MCOUNT[V9..V11,5]</p>
@MCPK[V##..V##,nn]	<p>Returns the Moving Capability Clearance for variable V## back nn historical points.</p> <p>The minimum value for historical points is 2.</p> <p>Ex: @MCPK[V9..V11,5]</p>
@MCR[V##..V##,nn]	<p>Returns the Moving Capability Ratio for variable V## back nn historical points.</p> <p>The minimum value for historical points is 2.</p> <p>Ex: @MCR[V9..V11,5]</p>
@MIN[V##..V##]	<p>Returns the minimum or smallest value from the variables in the list.</p> <p>Ex: @MIN[V15,V18,V21]</p>
@MMAX[V##..V##,nn]	<p>Returns the maximum or highest value from the variables in the list.</p> <p>The minimum value for historical points is 2.</p> <p>Ex: @MAX[V9..V11,5]</p>
@MMIN[V##..V##,nn]	<p>Returns the minimum or smallest value for variable V## back nn historical points.</p> <p>The minimum value for historical points is 2.</p> <p>Ex: @MMIN[V9..V11,5]</p>
@MO%OOL[V##..V##,nn]	<p>Returns the percentage of values observed to be out side of the control limits for variable V## back nn historical points.</p> <p>The minimum value for historical points is 2.</p> <p>Ex: @MO%OOL[V9..V11,5]</p>
@MOD[X,Y]	<p>Returns the integer remainder of the division X divided by Y.</p> <p>X and Y can be numbers (4) or variables V##.</p> <p>Ex: @MOD[V5,4]</p> <p>where V5 has a value of 19 returns a value of 3 (19 divided by 4 remainder 3).</p>
@MPREV[V##,nn]	<p>Returns the first non-null value for variable V## searching back nn historical points. The search starts with the last or previous record and then continues searching back ## - 1 historical points.</p> <p>The minimum value for historical points is 2.</p>

Calculated Variables using Formulas' and @Functions

	Ex: MPREV[V16,5]
@MRANGE[V##..V##,nn]	Returns the Moving Range for variable V## back nn historical points (Maximum - Minimum value). The minimum value for historical points is 2. Ex: MRANGE[V9..V11,5]
@MSD[V##..V##,nn]	Returns the Moving Standard Deviation for variable V## back nn historical points. The minimum value for historical points is 2. Ex: @MSD[V9..V11,5]
@MSUM[V##..V##,nn]	Returns the total of all values for variable V## back nn historical points. The minimum value for historical points is 2. Ex: @MSUM[V9..V11,5]
@MTZ[V##..V##,nn]	Returns the Moving Target Z values for variable V## back nn historical points. The minimum value for historical points is 2. Ex: @MTZ[V9..V11,5]
@NA[]	Returns a value of null. Ex: @IF[V5>99,@NA[],V5] if V5 is greater than 99 then the value null, @NA[], is assigned.
@OCTRL[V##..V##]	Returns the count of variables in the list whose values are found to exceed the control limits set for each variable in the list. Ex: @OCTRL[V5,V7..V14]
@OCTRL %[V##..V##]	Returns the percentage of variables in the list whose values are found to exceed the control limits set for each variable in the list. Ex: @OCTRL %[V5,V7..V14]
@OSPEC[V##..V##]	Returns the count of variables in the list whose values are found to exceed the specification limits set for each variable in the list. Ex: @OSPEC[V5..V9,V14]
@OSPEC %[V##..V##]	Returns the percentage of variables in the list whose values are found to exceed the specification limits set for each variable in the list. Ex: @OSPEC %[V5..V9,V14]
@PI[]	Returns the constant 3.1415926.

Calculated Variables using Formulas' and @Functions

	Ex: V18 * @PI[]
@POP[V##..V##,nn]	Returns statistics for the population defined in the range of variable back nn records. The use of the @POP function in a Calculated variable will trigger QW to display data on the Control Chart differently than other Calculated, Numeric Average or Range type variable types. Charting an @POP variable will show a vertical bar with the maximum value of the sample size at the top, the minimum value of the sample size at the bottom and the average as a blue dot. Ex: @POP[V9..V11,5]
@RANGE[V##..V##]	Returns the range (Maximum value - Minimum value) of the variables in the list. Ex: @RANGE[V4..V6,V9]
@RULE[V##..V##]	Returns the count of rule violations for the variables in the list. Ex: @RULE[V14..V18]
@RULE%[V##..V##]	Returns the percentage of variables in the list that had rule violations. Ex: @RULE%[V14..V18]
@SD[V##..V##]	Returns the Standard Deviation for the range of variables in the list Ex: @SD[V12,V17..V22]
@SGN[expression]	Determines the sign of the expression. (-1 for a negative, 0 for zero, 1 for a positive value) Ex: @SGN[V5-12]
@SIN[V##]	Returns the SINE of angle V## given in radians. Ex: @SIN[V13]
@SQRT[V##]	Returns the Square Root value. Ex: @SQRT[V16] (using V## * .5 returns the same value)
@SUM[V##..V##]	Returns the sum or total of adding together the variables in the list. Ex: @SUM[V15..V23]

Calculated Variables using Formulas' and @Functions

@TAN[V##]	Returns the TANGENT of angle V## given in radians. Ex: @TAN[V13]
@TARGET[V##]	Returns the Target value for a variable. Ex: @TARGET[V16]
Time Functions	Returns the current record hour in 24 hour clock format.
@HOUR[]	Returns the current record minute.
@MINUTE[]	Returns the current record second.
@SECOND[]	Ex: If the time on the current record is 2:10:03 PM @HOUR[] would return 14 @MINUTE[] would return 10 @SECOND[] would return 3
@TIMEDIFF[]	Returns the number of seconds since the previous record. Ex: @TIMEDIFF[] Use @TIMEDIFF[]/60 to derive minutes Use @TIMEDIFF[]/3600 to derive hours.
@UCL[V##]	Returns the Upper Control Limit value for a variable. Ex: @UCL[V16]
@USL[V##]	Returns the Upper Specification Limit value for a variable. Ex: @USL[V16]
@XBAR[V##..V##]	Returns statistics for the population defined in the range of variables. The use of the @XBAR function in a Calculated variable will trigger QW to display data on the Control Chart differently than other Calculated, Numeric Average or Range type variable types. Charting an @XBAR variable will show the average and the Range chart will be displayed below the Control Chart. The @XBAR function differs from the @AVG function by taking the sample size into consideration. Ex: @POP[V9..V11,5]