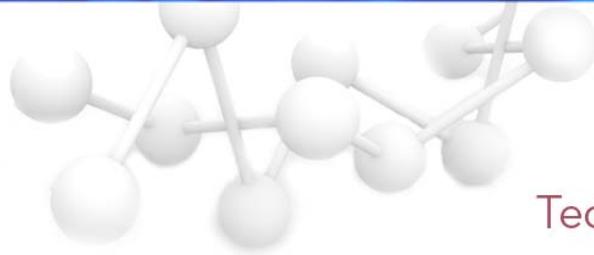


Errata Notice

This document contains references to BioTek. Please note that BioTek is now Agilent. This document is provided as a courtesy and is no longer kept current. For more information, go to www.agilent.com/lifescience/biotek.



Tech Note

Programming Long Equations into the BioTek™ Microplate Reader Family On-Board Software

The maximum length of a formula on the BioTek Microplate Reader family is 24 characters. For example, the token **NC;x** is counted as four individual characters, and the relational operator **<=** is counted as two individual characters by the internal software.

When translating kit insert instructions into a formula there are some hints that can help pare down long equations:

1. Where possible, leave parentheses out. The reader software will automatically perform results of an equation in order of operator priority. A multiplication or division operator (* or /) is higher priority than an addition or subtraction operator (+ or -), so it would be performed first regardless of its placement in the formula before an AND or OR token.

For example:

NC > NC;x - (.10*NC;x) AND NC < 0.100

(29 characters. Please note, spaces are provided for clarification only, they are not counted as part of the equation length)

can be rewritten as:

NC > NC;x - .1 * NC;x AND NC < .1 *(23 characters)*

*Note: If using a defined function in a formula, such as LOG, parentheses must be placed following the function to define on which well, well type, or number the function is performed; for example: **LOG(STD6;x)**.*

2. Omit leading and trailing zeroes. The number 0.250 can be entered as .25 with the same result.

For example:

PC;x > NC;x + 0.250 AND PC;x < 2.500 *(28 characters)*

can be rewritten as:

PC;x > NC;x + .25 AND PC;x < 2.5 *(24 characters)*

3. Use the shortest control identifier on the plate map. For example, if HPC or LPC can be used instead of CTL1 or CTL2 to identify a control on the map, then one less character would be counted in the formula.
4. Use the shortest mathematical route to the final result. For example, typing $.25 * NC;x$ is the same as typing $25\% NC;x$, yet uses one more character for the decimal point.
5. If a standard or control is validated on a range or percent of another control or standard then the equation may be able to be divided into two separate equations. However, if a standard or control is validated on an OD range or a numeric percentage of its own mean, then the equation can't be divided into two separate equations.

For example:

$$\mathbf{STD1 \geq STD0;x + .25 \text{ AND } STD1 \leq STD2;x - .25} \quad (35 \text{ characters})$$

can be rewritten as:

$$\mathbf{STD1 \geq STD0;x + .25} \quad (\text{first formula, 16 characters})$$

$$\mathbf{STD1 \leq STD2;x - .25} \quad (\text{second formula, 16 characters})$$

But, the criteria:

$$\mathbf{NC \geq NC;x - 25 \% NC;x \text{ AND } NC \leq NC;x + 25 \% NC;x} \quad (37 \text{ characters})$$

cannot be rewritten as:

$$\mathbf{NC \geq NC;x - 25\% NC;x} \quad (\text{first formula, 16 characters})$$

$$\mathbf{NC \leq NC;x + 25\% NC;x} \quad (\text{second formula, 16 characters})$$

The reason this example can't be rewritten as two separate equations is because the software logic automatically recalculates the control mean after each individual equation using only valid controls. Therefore, if any of the controls were found invalid and discarded according to the criteria of the first half of the formula then the recalculated mean used in the second formula will not be the same as in the first. The controls are not, then, validated against the same mean. Therefore, the possibility of dividing an equation longer than 24 characters into two separate formulae should be carefully analyzed before implementing. The following text helps illustrate this further.

Many clinical assays require a step in the 'Interpretation of Results' or 'Quality Control' section of the kit insert to check that a blank, control or standard meets a certain range of validity. Sometimes this includes checking that individual blanks, controls, or standards fall within a certain range of their absorbance mean.

An example of this might be criteria that states something like the following:

"Each negative control should be within +/- 25% of the mean of all negative controls. Discard any negative control that doesn't meet this criteria and recalculate the mean. Two of the three Negative Controls must be valid for the run to be valid."

This criteria translates into an equation like this:

$$\mathbf{NC < NC;x + (25 \% NC;x) \text{ AND } NC > NC;x - (25 \% NC;x)}$$

Generally, validation steps like this passing or failing is critical to accepting the entire results of the run. Many times the resulting control mean following this step is used in cutoffs, or other critical equations, that determine whether samples on the plate are positive or negative.

For these important reasons it becomes mandatory to program these instructions correctly into the instrument. After parentheses are removed from the equation above, there are still more than 24 characters. If this equation was divided into two separate equations (the first check NCs less than or equal to 25% of the mean, and the second to check NCs greater than or equal to 25% of the mean) it will give incorrect interpretation of results. Why? After each formula is performed the internal software discards any well types that don't meet the criteria and recalculates the mean using the valid well types only. This is done before any additional equations are performed. Therefore, in the above example, the second equation will use a mean different than the first equation if any of the NCs were determined to be invalid against the first set of criteria.

See the example below for more details:

There are three Negative Controls. The kit insert instructions and equation for validating the Negative Controls are as above. After the plate read the ODs are as follows:

First replicate NC = 0.050 OD
Second replicate NC = 0.070 OD
Third replicate NC = 0.040
NC mean = 0.0533
25 % NC mean = 0.0133
NC mean + 25 % NC mean = 0.0666
NC mean – 25 % NC mean = 0.040

If the equation were divided into two parts the first equation would be:

$$\text{NC} < \text{NC};x + 25\% \text{NC};x$$

Plugging in the given numbers would produce:

$$\text{NC} < 0.0533 + (0.0133) \text{ or} \\ \text{NC} < 0.066$$

In this case the NC with the OD of 0.070 would be discarded. The software would then recalculate the NC mean before proceeding with the next equation. The new NC mean would be 0.045 (the mean of the two remaining valid controls). 25% of that new mean is 0.01125. The second equation would be performed with the following numbers:

$$\text{NC} > \text{NC};x - 25\% \text{NC};x \text{ or} \\ \text{NC} > 0.045 - 0.01125 \text{ or} \\ \text{NC} > 0.03375$$

Since both the remaining NCs meet the criteria the assay would be valid because two of the three NCs remain valid. This is incorrect! In fact, the assay should be considered invalid because two of the three NCs are actually invalid, but because the equation was split in two the recalculated mean following the first equation weighted to the lower NC and it came into the valid range. The final mean of the NCs used in any equations following the validation step would be incorrect also. It should be noted that regardless of the order that the two separate equations

were input, the same result would occur, only the 0.040 NC would be discarded on the first equation, and the recalculated mean would have weighted higher so that the 0.070 NC would be valid.

Plugging the example numbers into the truly correct equation yields the following:

$$\begin{aligned} & \mathbf{NC < NC;x + 25\% NC;x \text{ AND } NC > NC;x - 25\% NC;x} \\ & \mathbf{NC < 0.0533 + 0.0133 \text{ AND } NC > 0.0533 - 0.0133} \\ & \mathbf{NC < 0.066 \text{ AND } NC > 0.040} \end{aligned}$$

In this case the two replicates outside the range would be invalidated, the final NC mean would correctly calculate to 0.050, and a warning message would appear on the report invalidating the results.

Therefore, when a control, blank, or standard is validated against criteria that includes its **own** mean, and the criteria is an equation longer than 24 characters, it is strongly recommended that **all** calculations using these well types be done manually to insure correct results.

If, however, a control, blank, or standard is validated against criteria including another well type mean, then the equation can be safely divided if it is longer than 24 characters.

For example, a kit insert may state:

“Each negative control should be greater than or equal to the mean of the blank well plus 0.050 and less than or equal to the mean of the positive controls minus 0.250.”

The equation would be illustrated as:

$$\mathbf{NC \geq blk;x + .05 \text{ AND } NC \leq PC;x - .25 \text{ (28 characters)}}$$

This equation can be divided into two separate equations because the negative controls are not validated against their own mean:

$$\begin{aligned} & \mathbf{NC \geq blk;x + .05 \text{ (first formula, 13 characters)}} \\ & \mathbf{NC \leq PC;x - .25 \text{ (second formula, 12 characters)}} \end{aligned}$$

Therefore, if any negative control were found invalid and discarded in one equation it would have no mathematical effect on determination of remaining controls against the second criteria.

In summary, some hints can be applied to equations to shorten their length. If the equation is still longer than 24 characters it may be divided into two separate equations if the blank, control, or standards aren't validated against their own mean. Otherwise, that criteria and any other criteria that then uses the mean of that well type should be manually calculated.

Rev. 5/12/05