Looping Structures

Function: A set of statements that perform a specific task and then return a value.

Built-in Function: A function provided by Visual Basic.

Condition: A Boolean expression that is tested for a "True" or "False" value.

Iteration: Each execution of a loop.

Pre-Test: The condition of the loop is tested before the loop executes.

Post-Test: The condition of the loop is tested after the loop executes.

Infinite Loop: A loop that never stops; the result of a logic error.

Procedural Scope: Can only be accessed by statements within the structure.

Step: An optional part of the For…Next structure that changes the way the counter is incremented (or decremented).

Sentinel (Flag): A condition used to stop a loop from executing.

Do…Loops

The Do…Loop structure uses a condition to determine when the looping should stop. The loop will continue to iterate until the condition is false; so "while" the condition is true, the loop continues iterating. The Do…Loop structure has two options for checking the condition: "Before" or "After" the statement(s) of the loop execute.

If there is a plausible scenario where you would not want the statement(s) of your loop to execute then the condition of your loop should be checked before the statement(s). This is referred to as a "Pre-Test". Pre-Testing your condition is accomplished using the Do While … Loop.

However, if you are sure that you will always want your loop to execute at least once then the condition of your loop should be checked after the statement(s). This is referred to as a "Post-Test". Post-Testing your condition is accomplished using the Do … Loop While.

Whether it is the Do While … Loop or the Do … Loop While, this is the structure you want to use when you, as the programmer, do not know how many times it will be necessary for the statement(s) to iterate.

For…Next Loop

The For…Next structure is used when you, as the programmer, actually know (or can calculate) the exact number of times your statement(s) will need to iterate. In place of a condition, this structure uses a counter.

The counter used in the For…Next loop is initialized, tested, and incremented within the loop. Remember, when a variable (in this case a counter variable) is initialized, it is declared ("created") and assigned a specific value at the same time. Being that the counter variable is created within this looping structure, the scope will be "Procedural"; which means it is only seen by the statement(s) within the loop. The lifetime of a Procedural scope lasts beyond the duration of not only the structure but also the procedure itself. If the Event procedure is ever triggered again, the counter will be reinitialized, thereby losing any previous value and starting over.

The counter is incremented by a value of 1 unless you, as the programmer, specify a different value using the optional "Step Value" with the keyword "Step". The step value you specify can be either a positive or a negative number.