




## How to Use the While Structure Tutorial

Functions -> Programming -> Structures  -> While Loop 

The While Loop structure in LabView is the same as a do-while loop that is used in other languages and requires a termination condition. The while loop is a structure that will continuously execute the functions inside it until the conditions for termination are reached. However, because the while structure is do-while, the termination condition is checked after at least one iteration has been completed.

To make the While Loop structure, first open the Functions palette and select the Programming palette. Under the Programming palette you will find the Structures palette . Open the

Structures palette and select the While Loop structure . When you click on the While Loop structure in the palette, you will notice that your cursor will turn into box with a circular arrow.

To create the While Loop structure, click and drag a box around the functions that you want to be in the While Loop. If you haven't defined or constructed these functions yet, just make a rectangle for how much space you think you'll need. You can always resize the While Loop structure later by dragging the handles.

Inside the While Loop structure you just created, there will be two functions already created for you. These are the Iteration Count (the blue box with the "i" in it) and the Loop Condition (the green box with a red dot in it).

Let's look at the following example which times how many seconds have elapsed and stops once it reaches 1 hours (or 3600 seconds). For the purposes of this example, we're going to assume that the other threads in this loop are virtually instantaneous compared to the Wait function.

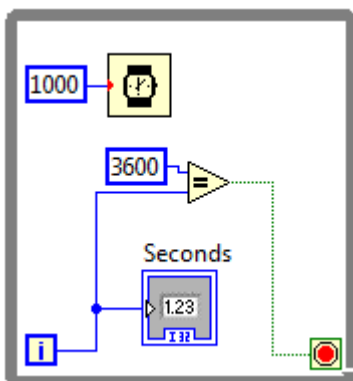




Figure 1

The blue box is an I32 data type that keeps track of how many iterations the loop has completed and is updated continuously throughout the execution (initialized to 0).

The green box is the Loop Condition and determines whether the loop should terminate when supplied a True or False value depending on the setting. The setting of the loop can be changed by clicking on the Loop Condition when it is not selected.

The Loop Condition takes in a Boolean value which indicates whether the termination conditions have been satisfied or not. When we want to use a setting that terminates when a True value is supplied to the Loop Condition, we use the setting with the red dot  which is equivalent to “stop if you get a True value”. If the loop has not satisfied the conditions, a False value should be supplied to the Loop Condition. (You will notice that in the Timer example that we stop once we receive a True value so we use the red dot option).

Likewise, when we use the alternate setting with a green circular arrow , the loop will terminate when supplied a False value and is equivalent to “stop when you get a False value”.

As with the case with all loops, the main pitfall with a structure like this is the possibility of having an infinite loop (when the termination conditions are never satisfied and the loop runs forever). If your VI does not behave the way you intended it to, there may be a problem with your loop termination conditions or how you calculate the termination conditions. There are also many other possible hazards associated with the while loop which are not discussed here.