

Data Propagation

Handling Propagation Problems

Note Within a loop, when an object with multiple outputs, such as the `DeMultiplexer`, sends data from one output on each loop iteration the other output values are invalidated at the beginning of each loop. This prevents propagating possibly old, incorrect data to the next object. This is also true for an object with multiple inputs, such as the `Build Record`.

When one input receives data on each loop iteration, values on all of its other inputs are invalidated at the beginning of each loop. VEE works this way to prevent a program from working with previous rather than current values, which can cause incorrect results.

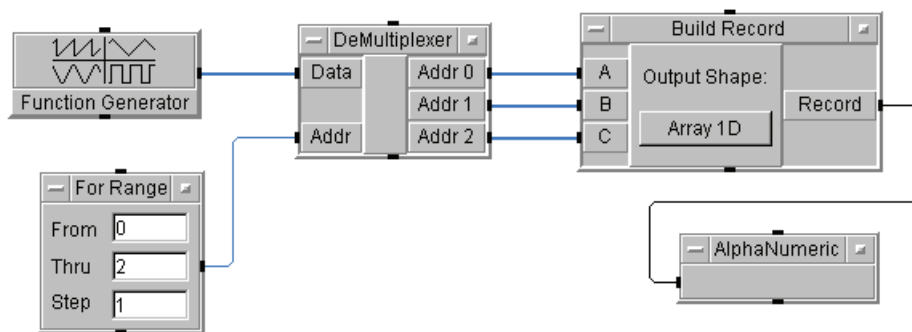


Figure 8-26. Invalid Data Inputs Stops Propagation on Build Record in a Loop

The `Build Record` object never propagates its `Record` output due to the way VEE loops work. In this program, each `For Range` iteration invalidates the data put on the `Build Record`'s inputs from the previous iteration. Since the `Build Record` object receives only one input on each loop iteration, only one input is valid at a time so there is no `Record` output.

Build this program yourself and turn on `Show Data Flow` to see how the `DeMultiplexer` only propagates one data output each time through the loop.

The program in Figure 8-27 makes this solution work as expected by using a `UserObject`. This solution works because a `UserObject`'s output terminals hold the data until the iterations are done. The data are valid on the `Build Record` inputs since the `UserObject` sends the three outputs at the same