

## Runtime Lookup Failure

At runtime, it is possible that a NVAR, SVAR or WAVE statement may fail. For example,

```
NVAR v1 = var1
```

will fail if `var1` does not exist in the current data folder when the statement is executed. You can use the `NVAR_Exists`, `SVAR_Exists`, and `WaveExists` functions to test if a given global reference is valid:

```
Function Test()
    NVAR/Z v1 = var1
    if (NVAR_Exists(v1))
        <use v1>
    endif
End
```

The `/Z` flag is necessary to prevent an error if the NVAR statement fails. You can also use it with SVAR and WAVE.

A common cause for failure is putting a WAVE statement in the wrong place. For example:

```
Function BadExample()
    WAVE w = resultWave
    <Call a function that creates a wave named resultWave>
    Display w
End
```

This function will compile successfully but will fail at runtime. The reason is that the `WAVE w = resultWave` statement has the runtime behavior of associating the local name `w` with a particular wave. But that wave does not exist until the following statement is executed. The function should be rewritten as:

```
Function GoodExample()
    <Call a function that creates a wave named resultWave>
    WAVE w = resultWave
    Display w
End
```

## Runtime Lookup Failure and the Debugger

You can break whenever a runtime lookup fails using the symbolic debugger (described in Chapter IV-8, **Debugging**). It is a good idea to do this, because it lets you know about runtime lookup failures at the moment they occur.

Sometimes you may create a WAVE, NVAR or SVAR reference knowing that the referenced global may not exist at runtime. Here is a trivial example:

```
Function Test()
    WAVE w = testWave
    if (WaveExists(testWave))
        Printf "testWave had %d points.\r", numpts(testWave)
    endif
End
```

If you enable the debugger's WAVE checking and if you execute the function when `testWave` does not exist, the debugger will break and flag that the WAVE reference failed. But you wrote the function to handle this situation, so the debugger break is not helpful in this case.

The solution is to rewrite the function using `WAVE/Z` instead of just `WAVE`. The `/Z` flag specifies that you know that the runtime lookup may fail and that you don't want to break if it does. You can use `NVAR/Z` and `SVAR/Z` in a similar fashion.