```
   ModifyGraph mode($t) = 3
End
```

Compile Time Versus Runtime

Because Igor user-defined functions are compiled, errors can occur during compilation (“compile time”) or when a function executes (“runtime”). It helps in programming if you have a clear understanding of what these terms mean.

Compile time is when Igor analyzes the text of all functions and produces low-level instructions that can be executed quickly later. This happens when you modify a procedure window and then:

- Choose Compile from the Macros menu.
- Click the Compile button at the bottom of a procedure window.
- Activate a window other than a procedure or help window.

Runtime is when Igor actually executes a function’s low-level instructions. This happens when:

- You invoke the function from the command line.
- The function is invoked from another procedure.
- Igor updates a dependency which calls the function.
- You use a button or other control that calls the function.

Conditions that exist at compile time are different from those at runtime. For example, a function can reference a global variable. The global does not need to exist at compile time, but it does need to exist at runtime. This issue is discussed in detail in the following sections.

Here is an example illustrating the distinction between compile time and runtime:

```
Function Example(w)
   WAVE w

   w = sin(x)
   FFT w
   w = r2polar(w)
End
```

The declaration “`WAVE w`” specifies that w is expected to be a real wave. This is correct until the FFT executes and thus the first wave assignment produces the correct result. After the FFT is executed at runtime, however, the wave becomes complex. The Igor compiler does not know this and so it compiled the second wave assignment on the assumption that w is real. A compile-time error will be generated complaining that r2polar is not available for this number type — i.e., real. To provide Igor with the information that the wave is complex after the FFT you need to rewrite the function like this:

```
Function Example(w)
   WAVE w

   w= sin(x)
   FFT w
   WAVE/C wc = w
   wc = r2polar(wc)
End
```

A statement like “`WAVE/C wc = w`” has the compile-time behavior of creating a symbol, wc, and specifying that it refers to a complex wave. It has the runtime behavior of making wc refer to a specific wave. The runtime behavior can not occur at compile time because the wave may not exist at compile time.