The compiler also needs to know if the wave is real, complex or text. Use Wave/C to create a complex wave reference and Wave/T to create a text wave reference. Wave by itself creates a real wave reference.

At runtime the Wave statement stores a reference to a specific wave in the wave reference variable (wOut in this example). The referenced wave must already exist when the wave statement executes. Otherwise Igor stores a NULL reference in the wave reference variable and you get an error when you attempt to use it. We put the `Wave wOut = $newName` statement *after* the Duplicate operation to insure that the wave exists when the Wave statement is executed. Putting the Wave statement before the command that creates the wave is a common error.

Automatic Creation of WAVE References

The Igor compiler sometimes automatically creates WAVE references. For example:

```
Function Example1()
   Make wave1
   wave1 = x^2
End
```

In this example, we did not declare a wave reference, and yet Igor was still able to compile an assignment statement referring to a wave. This is a feature of the **Make** operation (see page V-526) which automatically creates local references for simple object names. The **Duplicate** operation (see page V-185) and many other operations that create output waves also automatically create local wave references for simple object names.

Simple object names are names which are known at compile time for objects which will be created in the current data folder at runtime. Make and Duplicate do not create references if you use $<name>, a partial data folder path, or a full data folder path to specify the object.

In the case of Make and Duplicate with simple object names, the type of the automatically created wave reference is determined by flags. Make/C and Duplicate/C create complex wave references. Make/T and Duplicate/T create text wave references. Make and Duplicate without type flags create real wave references. See **WAVE Reference Types** on page IV-73 and **WAVE Reference Type Flags** on page IV-74 for a complete list of type flags and further details.

Most built-in operations that create output waves (often called "destination" waves) also automatically create wave references. For example, if you write:

```
DWT srcWave, destWave
```

it is as if you wrote:

```
DWT srcWave, destWave
WAVE destWave
```

After the discrete wavelet transform executes, you can reference `destWave` without an explicit wave reference.

Standalone WAVE Reference Statements

In cases where Igor does not automatically create a wave reference, because the output wave is not specified using a simple object name, you need to explicitly create a wave reference if you want to access the wave in an assignment statement.

You can create an explicit standalone wave reference using a statement following the command that created the output wave. In this example, the name of the output wave is specified as a parameter and therefore we can not use a simple object name when calling Make:

```
Function Example2(nameForOutputWave)
   String nameForOutputWave   // String contains the name of the wave to make

   Make $nameForOutputWave          // Make a wave
   Wave w = $nameForOutputWave      // Make a wave reference
   w = x^2
End
```