- From the Macros, Windows or user-defined menus
- From another macro
- From a button or other user control

The menu in which a macro appears, if any, is determined by the macro's type and subtype.

This table shows how a macro's type determines the menu that Igor puts it in.

Macro Type	Defining Keyword	Menu
Macro	Macro	Macros menu
Window Macro	Window	Windows menu
Proc	Proc	—

If a macro has a subtype, it may appear in a different menu. This is described under **Procedure Subtypes** on page IV-204. You can put macros in other menus as described in Chapter IV-5, **User-Defined Menus**.

You can not directly invoke a macro from a user function. You can invoke it indirectly, using the **Execute** operation (see page V-204).

Using $ in Macros

As shown in the following example, the $ operator can create references to global numeric and string variables as well as to waves.

```
Macro MacroTest(vStr, sStr, wStr)
   String vStr, sStr, wStr

   $vStr += 1
   $sStr += "Hello"
   $wStr += 1
End

Variable/G gVar = 0; String/G gStr = ""; Make/O/N=5 gWave = p
MacroTest("gVar", "gStr", "gWave")
```

See **String Substitution Using $** on page IV-18 for additional examples using $.

Waves as Parameters in Macros

The only way to pass a wave to a macro is to pass the name of the wave in a string parameter. You then use the $ operator to convert the string into a wave reference. For example:

```
Macro PrintWaveStdDev(w)
   String w

   WaveStats/Q $w
   Print V_sdev
End

Make/O/N=100 test=gnoise(1)
Print NamedWaveStdDev("test")
```

The Missing Parameter Dialog

When a macro that is declared to take a set of input parameters is executed with some or all of the parameters missing, it displays a dialog in which the user can enter the missing values. For example:

```
Macro MacCalcDiag(x,y)
   Variable x=10
```