How Waves Are Passed

Here is an example of a function "passing a wave" to a subroutine.

```
Function Routine()
   Make/O wave0 = x
   Subroutine(wave0)
End

Function Subroutine(w)
   WAVE w

   w = 1234    // This line DOES AFFECT the wave referred to by w.
End
```

We are really not passing a wave to Subroutine, but rather we are passing a reference to a wave. The parameter `w` is the wave reference.

Waves are global objects that exist independent of functions. The subroutine can use the wave reference to modify the contents of the wave. Using the terminology of "pass-by-value" and "pass-by-reference", the wave reference is passed by value, but this has the effect of "passing the wave" by reference.

In Igor Pro 8.00 or later, you can specify that a WAVE reference variable itself be passed by reference. This allows you to change the WAVE reference variable so that it references a different wave. For example:

```
Function Routine()
   Make/O wave0 = {1, 2, 3}
   WAVE w = wave0
   Print "Original:", w
   Subroutine(w)
   Print "New:", w
End

Function Subroutine(WAVE& ww)
   Make/O wave1 = {5, 6}
   WAVE ww = wave1// Changes w variable in Routine to reference wave1
End
```

Executing Routine prints:

```
Original:
wave0[0]= {1,2,3}
  New:
wave1[0]= {5,6}
```

Using Optional Parameters

Following is an example of a function with two optional input parameters:

```
Function OptionalParameterDemo(a,b, [c,d])
   Variable a,b,c,d           // c and d are optional parameters

   // Determine if the optional parameters were supplied
   if( ParamIsDefault(c) && ParamIsDefault(d) )
      Print "Missing optional parameters c and d."
   endif

   Print a,b,c,d
End
```

Executing on the command line with none of the optional inputs:

```
OptionalParameterDemo(8,6)
  Missing optional parameters c and d.
  8  6  0  0
```