Integer Parameters

In Igor Pro 7 or later you can use these integer types for parameters and local variables in user-defined functions:

`int`	32-bit integer in IGOR32; 64-bit integer in IGOR64
`int64`	64-bit signed integer
`uint64`	64-bit unsigned integer

`int` is a generic signed integer that you can use for wave indexing or general use. It provides a speed improvement over Variable or double in most cases.

Signed `int64` and unsigned `uint64` are for special purposes where you need explicit access to bits. You can also use them in structures.

Optional Parameters

Following the list of required function input parameters, you can also specify a list of optional input parameters by enclosing the parameter names in brackets. You can supply any number of optional parameter values when calling the function by using the *ParamName=Value* syntax. Optional parameters may be of any valid data type. There is no limit on the number of parameters.

All optional parameters must be declared immediately after the function declaration. As with all other variables, optional parameters are initialized to zero. You must use the **ParamIsDefault** function to determine if a particular optional parameter was supplied in the function call.

See **Using Optional Parameters** on page IV-60 for an example.

Inline Parameters

In Igor Pro 7 or later you can declare user-defined functions parameters inline. This means that the parameter types and parameter names are declared in the same statement as the function name:

```
Function Example(Variable a, [ Variable b, double c ])
   Print a,b,c
End
```

or, equivalently:

```
Function Example2(
      Variable a,// The comma is optional
      [
         Variable b,
         double c
      ]
   )
   Print a,b,c
End
```