

Here, “ : ButtonControl” identifies a macro intended to be called when a user-defined button control is clicked. Because of the subtype, this macro is added to the menu of procedures that appears in the Button Control dialog. When Igor automatically generates a procedure it generates the appropriate subtype. See **Procedure Subtypes** on page IV-204 for details.

## The Parameter List and Parameter Declarations

The parameter list specifies the name for each input parameter. Macros have a limit of 10 parameters.

The parameter declaration must declare the type of each parameter using the keywords `Variable` or `String`. If a parameter is a complex number, it must be declared `Variable/C`.

**Note:** There should be no blank lines or other commands until after all the input parameters are defined. There should be one blank line after the parameter declarations, before the rest of the procedure. Igor will report errors if these conditions are not met.

Variable and string parameters in macros are always passed to a subroutine by value.

When macros are invoked with some or all of their input parameters missing, Igor displays a missing parameter dialog to allow the user to enter those parameters. In the past this has been a reason to use macros. However, as of Igor Pro 4, functions can present a similar dialog to fetch input from the user, as explained under **The Simple Input Dialog** on page IV-144.

## Local Variable Declarations

The input parameter declarations are followed by the local variable declarations if the macro uses local variables. Local variables exist only during the execution of the macro. They can be numeric or string and are declared using the `Variable` or `String` keywords. They can optionally be initialized. Here is an example:

```
Macro Example(p1)
    Variable p1

    // Here are the local variables
    Variable v1, v2
    Variable v3=0
    Variable/C cv1=cplx(0,0)
    String s1="test", s2="test2"

    <Body code>
End
```

If you do not supply explicit initialization, Igor automatically initializes local numeric variables with the value zero and local string variables with the value "".

The name of a local variable is allowed to conflict with other names in Igor although they must be unique within the macro. Clearly if you create a local variable named “sin” then you will be unable to use Igor’s built-in sin function within the macro.

You can declare a local variable in any part of a macro with one exception. If you place a variable declaration inside a loop in a macro then the declaration will be executed multiple times and Igor will generate an error since local variable names must be unique.