

The last two lines of Subroutine set the value of the local variables v and s. They have no effect on the value of the variables v and s in the calling Routine. What is passed to Subroutine is the numeric value 4321 and the string value "Hello".

Pass-By-Reference

You can specify that a parameter to a function is to be passed by reference rather than by value. In this way, the function called can change the value of the parameter and update it in the calling function. This is much like using pointers to arguments in C++. This technique is needed and appropriate only when you need to return more than one value from a function.

Functions with pass-by-reference parameters can only be called from other functions — not from the command line.

Only numeric variables (declared by Variable, double, int, int64, uint64 or complex), string variables, structures, DFREF, and WAVE variables can be passed by reference. Structures are always passed by reference. Pass by reference DFREF and WAVE variables were added in Igor Pro 8.00.

The variable or string being passed must be a local variable and can not be a global variable. To designate a variable or string parameter for pass-by-reference, simply prepend an ampersand symbol before the name in the parameter declaration:

```
Function Subroutine(num1,num2,str1)
    Variable &num1, num2
    String &str1

    num1= 12+num2
    str1= "The number is"
End
```

and then call the function with the name of a local variable in the reference slot:

```
Function Routine()
    Variable num= 1000
    String str= "hello"
    Subroutine(num,2,str)
    Print str, num
End
```

When executed, Routine prints "The number is 14" rather than "hello 1000", which would be the case if pass-by-reference were not used.

A pass-by-reference parameter can be passed to another function that expects a reference:

```
Function SubSubroutine(b)
    Variable &b
    b= 123
End

Function Subroutine(a)
    Variable &a
    SubSubroutine(a)
End

Function Routine()
    Variable num
    Subroutine(num)
    print num
End
```

You can not pass NVARs, SVARS, or FUNCREFs by reference to a function. You can use a structure containing fields of these types to achieve the same end.