

```

STRUCT DemoStruct s
WAVE/T w
[ s, w ] = Subroutine(2)
Print s, w
End

```

Executing CallingRoutine gives:

```

STRUCT DemoStruct
num: 6
str: from Subroutine
tw[0]= {"text wave point 0","text wave point 1"}

```

In Igor Pro 9.00 and later, the CallingRoutine can be rewritten to declare the return variables in the destination parameter list:

```

Function CallingRoutine()
[ STRUCT DemoStruct s, WAVE/T w ] = Subroutine(2)
Print s, w
End

```

At runtime, WAVE variables declared in the destination list do not perform the usual time consuming lookup (trying to find a wave named w in this case.) They act as if WAVE/ZZ was specified. See **WAVE** on page V-1069 for details.

### Using Multiple Return Syntax Variables As Input and Output

Igor initializes local numeric variables to 0, local string variables to NULL, and local WAVE variables to NULL. If you use a local variable as both an input and an output to a function that uses multiple return syntax, you should assign a value to the variable before using it:

```

Function [String str] AppendToString()
str += "DEF"
End

Function CallingRoutineBad()
String str                      // Initialized by Igor to NULL
[str] = AppendToString()        // Error: Attempt to use a NULL string
Print str
End

Function CallingRoutineGood()
String str = "ABC"              // Explicitly initialized
[str] = AppendToString()
Print str                        // Prints "ABCDEF"
End

```

CallingRoutineBad passes a NULL string to AppendToString which treats it as both an input and an output. AppendToString attempts to append to the NULL string causing an "attempt to use NULL string" error.

Here is a more subtle way to make this mistake:

```

Function CallingRoutineBad()
[String str] = AppendToString() // Error: Attempt to use a NULL string
Print str
End

```

Again, str is NULL and this results in an "attempt to use NULL string" error.

The same principle applies to local numeric variables and to WAVE references. If you are passing them to an MRS function that uses them as both an input and an output, you must initialize them in the calling function.