A proc picture defined in a regular module is usually intended to be used in that module only but can also be used from a global procedure file using the qualified name. It can not be used from an independent module.

Proc Pictures in Independent Modules

Here is an example of a proc picture in an independent module:

```
#pragma IndependentModule = MyIndependentModule

Picture MyIndependentPicture
   ASCII85Begin
   ...
   ASCII85End
End
```

The static keyword is not used but the the picture name is still in the namespace of the independent module.

To draw a proc picture defined in an independent module you must qualify the picture name with the name of the independent module:

```
DrawPICT 0,0,1,1,MyIndependentModule#MyIndependentPicture
```

A proc picture defined in an independent module is usually intended to be used in that module only but can also be used from any procedure file using the qualified name.

How Parameters Work

There are two ways of passing parameters from a routine to a subroutine:

- Pass-by-value
- Pass-by-reference

"Pass-by-value" means that the routine passes the *value* of an expression to the subroutine. "Pass-by-reference" means that the routine passes *access to a variable* to the subroutine. The important difference is that, in pass-by-reference, the subroutine can change the original variable in the calling routine.

Like C++, Igor allows either method for numeric and string variables. You should use pass-by-value in most cases and reserve pass-by-reference for those situations in which you need multiple return values.

Example of Pass-By-Value

```
Function Routine()
   Variable v = 4321
   String s = "Hello"

   Subroutine(v, s)
End

Function Subroutine(v, s)
   Variable v
   String s

   Print v, s

   // These lines have NO EFFECT on the calling routine.
   v = 1234
   s = "Goodbye"
End
```

Note that v and s are *local variables* in Routine. In Subroutine, they are *parameters* which act very much like local variables. The names "v" and "s" are local to the respective functions. The v in Subroutine is not the same variable as the v in Routine although it initially has the same value.