See also: **Runtime Lookup of Globals** on page IV-65
Automatic Creation of WAVE References on page IV-72
Automatic Creation of WAVE, NVAR and SVAR References on page IV-70
Legacy Code Issues on page IV-113

The version Pragma

The version pragma sets the version of the procedure file. It is optional and is of interest mostly if you are the developer of a package used by a widespread group of users.

For details on the version pragma, see **Procedure File Version Information** on page IV-166.

The IgorVersion Pragma

The IgorVersion pragma is also optional and of interest to developers of packages. It gives you a way to prevent procedures from compiling under versions of Igor Pro older than the specified version number.

For example, the statement:

```
#pragma IgorVersion = 7.00
```

tells Igor that the procedure file requires Igor Pro 7.00 or later.

The hide Pragma

The hide pragma allows you to make a procedure file invisible.

For details on the hide pragma, see **Invisible Procedure Files** on page III-402.

The ModuleName Pragma

The ModuleName pragma gives you the ability to use static functions and proc pictures in a global context, such as in the action procedure of a control or on the command line. Using this pragma entails a two step process: define a name for the procedure file, and then use a special syntax to access objects in the named procedure file.

To define a module name for a procedure file use the format:

```
#pragma ModuleName = name
```

This statement associates the specified module name with the procedure file in which the statement appears.

You can then use objects from the named procedure file by preceding the object name with the name of the module and the # character. For example:

```
#pragma ModuleName = MyModule

Static Function Test(a)
   Variable a

   return a+100
End
```

Then, on the command line or from another procedure file, you can execute:

```
Print MyModule#Test(3)
```

Choose a distinctive module name so that it does not clash with module names used by other programmers. WaveMetrics uses module names with a `WM_` prefix, so you must not use such names.

For further discussion see **Regular Modules** on page IV-236.