To conditionally compile based on the version of Igor, use:

```
#if IgorVersion() < 7.00
   <code for Igor6 or before>
#else
   <code for Igor7 or later>
#endif
```

Conditional Compilation Directives

The conditional compiler directives are modeled after the C/C++ language. Unlike other #keyword directives, these may be indented. For defining symbols, the directives are:

```
#define symbol

#undef symbol
```

For conditional compilation, the directives are:

```
#ifdef symbol

#ifndef symbol

#if expression

#elif expression

#else

#endif
```

Expressions are ordinary Igor expressions, but cannot involve any user-defined objects. They evaluate to TRUE if the absolute value is > 0.5.

Conditionals must be either completely outside or completely inside function definitions; they cannot straddle a function definition. Conditionals cannot be used within macros but the **defined** function can.

Nesting depth is limited to 16 levels. Trailing text other than a comment is illegal.

Conditional Compilation Symbols

#define is used purely for defining symbols (there is nothing like C's preprocessor) and the only use of a symbol is with #if, #ifdef, #ifndef and the defined function.

The defined function allows you to test if a symbol was defined using #define:

```
#if defined(symbol)
```

Symbols exist only in the file where they are defined; the only exception is for symbols defined in the main procedure window, which are available to all other procedure files except independent modules. In addition, you can define global symbols that are available in all procedure windows (including independent modules) using:

```
SetIgorOption poundDefine=symb
```

This adds one symbol to a global list. You can query the global list using:

```
SetIgorOption poundDefine=symb?
```

This sets V_flag to 1 if symb exists or 0 otherwise. To remove a symbol from the global list use:

```
SetIgorOption poundUndefine=symb
```

For non-independent module procedure windows, a symbol is defined if it exists in the global list *or* in the main procedure window's list *or* in the given procedure window.

For independent module procedure windows, a symbol is defined if it exists in the global list *or* in the given procedure window; it does not use the main procedure window list.

A symbol defined in a global list is not undefined by a #undef in a procedure window.