

- Choose a clear and specific name for your utility routine.  
By choosing a name that says precisely what your utility routine does, you minimize the likelihood of collision with the name of another procedure. You also increase the readability of your program.
- Make functions which are used only internally by your procedures static.  
By making internal functions static (i.e., private), you minimize the likelihood of collision with the name of another procedure.

## Programming with Liberal Names

Standard names in Igor can contain letters, numbers and the underscore character only.

It is possible to use wave names and data folder names that contain almost any character. Such names are called “liberal names” (see **Liberal Object Names** on page III-501).

As this section explains, programmers need to use special techniques for their procedures to work with liberal names. Consequently, if you do use liberal names, some existing Igor procedures may break.

Whenever a liberal name is used in a command or expression, the name must be enclosed in single quotes. For example:

```
Make 'Wave 1', wave2, 'miles/hour'  
'Wave 1' = wave2 + 'miles/hour'
```

Without the single quotes, Igor has no way to know where a particular name ends. This is a problem whenever Igor parses a command or statement. Igor parses commands at the following times:

- When it compiles a user-defined function.
- When it compiles the right-hand side of an assignment statement, including a formula (:= dependency expression).
- When it interprets a macro.
- When it interprets a command that you enter in the command line or via an Igor Text file.
- When it interprets a command you submit for execution via the Execute operation.
- When it interprets a command that an XOP submits for execution via the XOPCommand or XOPSilentCommand callback routines.

When you use an Igor dialog to generate a command, Igor automatically uses quotes where necessary.

Programmers need to be concerned about liberal names whenever they create a command and then execute it, via an Igor Text file, the Execute operation or the XOPCommand and XOPSilentCommand callback routines, and when creating a formula. In short, when you create something that Igor has to parse, names must be quoted if they are liberal. Names that are not liberal can be quoted or unquoted.

If you have a procedure that builds up a command in a string variable and then executes it via the Execute operation, you must use the PossiblyQuoteName function to provide the quotes if needed.

Here is a trivial example showing the liberal-name unaware and liberal-name aware techniques:

```
Function AddOneToWave(w)  
    WAVE w  
  
    String cmd  
    String name = NameOfWave(w)  
  
    // Liberal-name unaware - generates error with liberal name  
    sprintf cmd, "%s += 1", name  
    Execute cmd  
  
    // Liberal-name aware way  
    sprintf cmd, "%s += 1", PossiblyQuoteName(name)
```