

### \$ Precedence Issues In Commands

There is one case in which string substitution does not work as you might expect. Consider this example:

```
String str1 = "wave1"
wave2 = $str1 + 3
```

(The uses of \$ in this section apply to the command line and macros only, not to user-defined functions in which you must use **Wave References** to read and write waves.)

You might expect that this would cause Igor to set wave2 equal to the sum of wave1 and 3. Instead, it generates an “expected string expression” error. The reason is that Igor tries to concatenate str1 and 3 *before* doing the substitution implied by \$. The + operator is also used to concatenate two string expressions, and it has higher precedence than the \$ operator. Since str1 is a string but 3 is not, Igor cannot do the concatenation.

You can fix this by changing this wave assignment to one of the following:

```
wave2 = 3 + $str1
wave2 = ($str1) + 3
```

Both of these accomplish the desired effect of setting wave2 equal to the sum of wave1 and 3. Similarly,

```
wave2 = $str1 + $str2 // Igor sees "$(str1 + str2)"
```

generates the same “expected string expression” error. The reason is that Igor is trying to concatenate str1 and \$str2. \$str2 is a name, not a string. The solution is:

```
wave2 = ($str1) + ($str2) // sets wave2 to sum of two named waves
```

Another situation arises when using the \$ operator and [. The [ symbol can be used for either point indexing into a wave, or byte indexing into a string. The commands

```
String wvName = "wave0"
$wvName[1,2] = wave1[p] // sets two values in wave named "wave0"
```

are interpreted to mean that points 1 and 2 of wave0 are set values from wave1.

If you intended “\$wvName[1,2] = wave1” to mean that a wave whose name comes from bytes 1 and 2 of the wvName string (“av”) has all of its values set from wave1, you must use parenthesis:

```
$(wvName[1,2]) = wave1 // sets all values of wave named "av"
```

### String Utility Functions

WaveMetrics provides a number of utility functions for dealing with strings. To see a list of the built-in string functions:

1. Open the Igor Help Browser Command Help tab.
2. Open the Advanced Filtering control.
3. Uncheck all checkboxes except for Functions.
4. Choose String from the Functions pop-up menu.

See **Character-by-Character Operations** on page IV-173 for an example of stepping through characters using user-defined functions.

### Special Cases

This section documents some techniques that were devised to handle certain specialized situations that arise with respect to Igor’s command language.