See also **Using Bitwise Operators** on page IV-42.

RGBA Values

This section explains RGBA values used to specify colors in commands other than Gizmo commands. For Gizmo commands, see **Gizmo Color Specification** on page II-428.

In commands, colors are specified as RGBA values in the form (r,g,b[,a]).

r, g, and b specify the amount of red, green and blue in the color as integers from 0 to 65535.

The optional parameter a specifies "alpha" which represents the opacity of the color as an integer from 0 (fully transparent) to 65535 (fully opaque). a defaults to 65535 (fully opaque).

(0,0,0) represents opaque black and (65535,65535,65535) represents opaque white.

For example:

```
ModifyGraph rgb(wave0)=(0,0,0)                // Opaque black
ModifyGraph rgb(wave0)=(65535,65535,65535)    // Opaque white
ModifyGraph rgb(wave0)=(65535,0,0,30000)      // Translucent red
```

Working With Strings

Igor has a rich repertoire of string handling capabilities. See **Strings** on page V-11 for a complete list of Igor string functions. Many of the techniques described in this section will be of interest only to programmers.

Many Igor operations require *string* parameters. For example, to label a graph axis, you can use the Label operation:

```
Label left, "Volts"
```

Other Igor operations, such as Make, require *names* as parameters:

```
Make wave1
```

Using the string substitution technique, described in **String Substitution Using $** on page IV-18, you can generate a name parameter by making a string containing the name and using the $ operator:

```
String stringContainingName = "wave1"
Make $stringContainingName
```

String Expressions

Wherever Igor requires a string parameter, you can use a string expression. A string expression can be:

- A literal string (`"Today is"`)
- The output of a string function (`date()`)
- An element of a text wave (`textWave0[3]`)
- A UTF-16 literal (`U+2022`)
- Some combination of string expressions (`"Today is" + date()`)

In addition, you can derive a string expression by indexing into another string expression. For example,

```
Print ("Today is" + date())[0,4]
```

prints “Today”.

A string variable can store the result of a string expression. For example:

```
String str1 = "Today is" + date()
```

A string variable can also be part of a string expression, as in:

```
Print "Hello. " + str1
```