Expressions as Parameters

In an operation, function, or macro parameter list which has a numeric parameter you can always use a numeric expression instead of a literal number. A numeric expression is a legal combination of literal numbers, numeric variables, numeric functions, and numeric operators. For example, consider the command

```
SetScale x, 0, 6.283185, "v", wave1
```

which sets the X scaling for wave1. You could also write this as

```
SetScale x, 0, 2*PI, "v", wave1
```

Literal numbers include hexadecimal literals introduced by "0x". For example:

```
Printf "The largest unsigned 16-bit number is: %d\r", 0xFFFF
```

Parentheses Required for /N=(<expression>)

Many operations accept flags of the form "/A=n" where A is some letter and n is some number. You can use a numeric expression for n but you must parenthesize the expression.

For example, both:

```
Make/N=512 wave1
Make/N=(2^9) wave1
```

are legal but this isn't:

```
Make/N=2^9 wave1
```

A variable name is a form of numeric expression. Thus, assuming v1 is the name of a variable:

```
Make/N=(v1)
```

is legal, but

```
Make/N=v1
```

is not. This parenthesization is required only when you use a numeric expression in an operation flag.

String Expressions

A string expression can be used where Igor expects a string parameter. A string expression is a legal combination of literal strings, string variables, string functions, UTF-16 literals, and the string operator + which concatenates strings.

A UTF-16 literal represents a Unicode code value and consists of "U+" followed by four hexadecimal digits. For example:

```
Print "This is a bullet character: " + U+2022
```

Setting Bit Parameters

A number of commands require that you specify a bit value to set certain parameters. In these instances you set a certain bit *number* by using a specific bit *value* in the command. The bit value is 2^n, where *n* is the bit number. So, to set bit 0 use a bit value of 1, to set bit 1 use a bit value of 2, etc.

For the example of the TraceNameList function the last parameter is a bit setting. To select normal traces you must set bit 0:

```
TraceNameList("",";",1)
```

and to select contour traces set bit 1:

```
TraceNameList("",";",2)
```

Most importantly, you can set multiple bits at one time by adding the bit values together. Thus, for TraceNameList you can select both normal (bit 0) and contour (bit 1) traces by using:

```
TraceNameList("",";",3)
```