Alphabetic Listing of Functions, Operations and Keywords

This section alphabetically lists all built-in functions, operations and keywords. Much of this information is also accessible online in the Command Help tab of the Igor Help Browser.

External operations (XOPs) and external functions (XFUNCs) are not covered here. For information about them, use the Command Help tab of the Igor Help Browser and the XOP help file in the same folder as the XOP file.

Reference Syntax Guide

In the descriptions of functions and operations that follow, italics indicate parameters for which you can supply numeric or string expressions. Non-italic keywords must be entered literally as they appear. Commas, slashes, braces and parentheses in these descriptions are always literals. Brackets surround optional flags or parameters. Ellipses (…) indicate that the preceding element may be repeated a number of times.

Italicized parameters represent values you supply. Italic words ending with "*Name*" are names (wave names, for example), and those ending with "*Str*" are strings. Italic words ending with "*Spec*" (meaning "specification") are usually further defined in the description. If none of these endings are employed, the italic word is a numeric expression, such as a literal number, the name of a variable or function, or some valid combination.

Strings and names are different, but you can use a string where a name is expected using "string substitution": precede a string expression with the $ operator. See **String Substitution Using $** on page IV-18.

A syntax description may span several lines, but the actual command you create must occupy a single line.

Many operations have optional "flags". Flags that accept a value (such as the Make operation's /N=*n* flag) sometimes require additional parentheses. For example:

```
Make/N=1 aNewWave
```

is acceptable because here *n* is the literal "1". To use a numeric expression (anything other than a literal number) for *n*, parentheses are needed:

```
Make/N=(numberOfPoints) aNewWave          // error if no parentheses!
```

For more about using functions, operations and keywords, see Chapter IV-1, **Working with Commands**, Chapter IV-2, **Programming Overview**, and Chapter IV-10, **Advanced Topics**.