

Overview

This chapter discusses the properties and uses of global numeric and string variables. For the fine points of programming with global variables, see [Accessing Global Variables and Waves](#) on page IV-65.

Numeric variables are double-precision floating point and can be real or complex. String variables can hold an arbitrary number of bytes. Igor stores all global variables when you save an experiment and restores them when you reopen the experiment.

Numeric variables or numeric expressions containing numeric variables can be used in any place where literal numbers are appropriate including as operands in assignment statements and as parameters to operations, functions, and macros.

When using numeric variables in operation flag parameters, you need parentheses. See [Reference Syntax Guide](#) on page V-15.

String variables or string expressions can be used in any place where strings are appropriate. String variables can also be used as parameters where Igor expects to find the name of an object such as a wave, variable, graph, table or page layout. For details on this see [Converting a String into a Reference Using \\$](#) on page IV-62.

In Igor7 or later, Igor assumes that the contents of string variables are encoded as UTF-8. If you store non-ASCII text in string variables created by Igor6 or before, you need to convert it for use in Igor7 or later. See [String Variables and Text Encodings](#) on page II-106 for details.

Creating Global Variables

There are 20 built-in numeric variables (K0 ... K19), called system variables, that exist all the time. Igor uses these mainly to return results from the CurveFit operation. We recommend that you refrain from using system variables for other purposes.

All other variables are user variables. User variables can be created in one of two ways:

- Automatically in the course of certain operations
- Explicitly by the user, via the Variable/G and String/G operations

When you create a variable directly from the command line using the Variable or String operation, it is always global and you can omit the /G flag. You need /G in Igor procedures to make variables global. The /G flag has a secondary effect — it permits you to overwrite existing global variables.

Uses For Global Variables

Global variables have two properties that make them useful: globalness and persistence. Since they are global, they can be accessed from any procedure. Since they are persistent, you can use them to store settings over time.

Using globals for their globalness creates non-explicit dependencies between procedures. This makes it difficult to understand and debug them. Using a global variable to pass information from one procedure to another when you could use a parameter is bad programming and should be avoided except under rare circumstances. Consequently, you should use global variables when you need persistence.

A legitimate use of a global variable for its globalness is when you have a value that rarely changes and needs to be accessed by many procedures.

Variable Names

Variable names consist of 1 to 255 bytes. Only ASCII characters are allowed. The first character must be alphabetic. The remaining characters can be alphabetic, numeric or the underscore character. Names in Igor are case insensitive.