WaveExists

`WaveExists(`*`wave`*`)`

The WaveExists function returns one if wave reference is valid, or zero if the wave reference is null. For example if, in a user function, you have:

```
Wave w= $"no such wave"
```

then `WaveExists(w)` will return zero.

Details

If wave is not a valid wave reference, WaveExists also checks whether wave itself is the name of an existing wave when executing the function. WaveExists should be used in functions only. In macros, use the exists function instead.

See Also

WAVE, **exists**, **NVAR_Exists**, **SVAR_Exists**, and **Accessing Global Variables and Waves** on page IV-65.

WaveHash

`WaveHash(`*`wave, method`*`)`

The WaveHash function returns a cryptographic hash of the bytes of the wave's data using the specified method. All contents of the wave's header are ignored.

Parameters

wave may reference a numeric or text wave.

method is a number indicating the hash algorithm to use:

1	SHA-256 (SHA-2)
2	MD4
3	MD5
4	SHA-1
5	SHA-224 (SHA-2)
6	SHA-384 (SHA-2)
7	SHA-512 (SHA-2)

See Also

WaveModCount, **WaveCRC**, **StringCRC**, **Hash**

WaveInfo

`WaveInfo(`*`waveName`*`, 0)`

The WaveInfo function returns a string containing a semicolon-separated list of information about the named wave.

The second parameter is reserved for future use and must be zero.

Details

The result string contains a list of keyword-value pairs. Each pair consists of a keyword, a colon, the value text, and a semicolon.

Here are the keywords:

Always pass 0 as the second input parameter. In future versions of Igor, this parameter may request other kinds of information to be returned.

A null wave reference returns a zero-length string. This might be encountered, for instance, when using **WaveRefIndexedDFR** in a loop to act on all waves in a data folder, and the loop has incremented beyond the highest valid index.

Keyword	Information Following Keyword
DUNITS	The wave’s data units.
FULLSCALE	Three numbers indicating whether the wave has any data full scale information, and the min and max data full scale values. The format of the FULLSCALE description is: `FULLSCALE:`*`validFS`*`,`*`minFS`*`,`*`maxFS`*`;` *validFS* is 1 if *minFS* and *maxFS* have been set via a `SetScale d` command; otherwise it is 0.
LOCK	Reads back the value set by **SetWaveLock**.
MODIFIED	1 if the wave has been modified since the experiment was last saved, else 0.
MODTIME	The date and time that the wave was last modified in seconds since January 1, 1904.
NUMTYPE	A number denoting the data type of the wave. For text waves this is 0. For wave reference waves it is 16384. For data folder reference waves it is 256. For numeric waves it is one of the following: 1: Complex, added to one of the following 2: 32-bit (single precision) floating point 4: 64-bit (double precision) floating point 8: 8-bit signed integer 16: 16-bit signed integer 32: 32-bit signed integer 128: 64-bit signed integer 64: Unsigned, added to 8, 16, 32 or 128 if wave is unsigned For example, the number denoting a complex double precision wave is 5 (i.e., 1+4).
PATH	The name of the symbolic path in which the wave file is stored (e.g., PATH:home;) or nothing if there is no path for the wave (PATH:;).
SIZEINBYTES	The total size of the wave in bytes. This includes the wave’s header, data, note, dimension labels, and unit strings. This keyword was added in Igor Pro 7.00.
XUNITS	The wave’s X units.

Examples

```
Make/O wave1;SetScale x,0,1,"dyn",wave1;SetScale y,3,20,"v",wave1
String info = WaveInfo(wave1,0)
Print NumberByKey("NUMTYPE", info)          // Prints 2
Print StringByKey("DUNITS", info)           // Prints "v"
```

See Also

The functions and operations listed under “About Waves” categories in the Command Help tab of the Igor Help Browser; among them are **CreationDate**, **ModDate**, **WaveModCount**, **WaveType**, **note**, and **numpnts**.

NumberByKey and **StringByKey** functions for parsing the returned keyword list.

WaveInfo lacks information about multidimensional waves. Individual functions are provided to return dimension-related information: **DimDelta**, **DimOffset**, **DimSize**, **WaveUnits**, and **GetDimLabel**.