WAVEClear ensures that memory is deallocated after waves are killed as in this example:

```
Function foo()
   Make wave1
   FunctionThatKillsWave1()
   WAVEClear wave1
   AnotherFunction()
End
```

Although memory used for wave1 will be deallocated when `foo` returns, that memory will not be automatically released while the function executes because the WAVE variable still contains a reference to the wave. In this example, WAVEClear deallocates that memory before `AnotherFunction` executes.

You can also use WAVEClear before passing a data folder to preemptive threads using **ThreadGroupPutDF**.

See Also

Accessing Waves in Functions on page IV-82, **Wave Reference Counting** on page IV-205, and **ThreadSafe Functions and Multitasking** on page IV-329.

WaveCRC

WaveCRC(*inCRC*, *waveName* [, *checkHeader*])

The WaveCRC function returns a 32-bit cyclic redundancy check value of the bytes in the named wave starting with *inCRC*.

Pass 0 for *inCRC* the first time you call WaveCRC for a particular stream of bytes as represented by the wave data.

Pass the last-returned value from WaveCRC for *inCRC* if you are creating a CRC value for a given stream of bytes through multiple calls to WaveCRC.

waveName may be a numeric or text wave.

The optional *checkHeader* parameter determines how much of the wave is checked:

checkHeader	What It Does
0	Check only the wave data (default).
1	Check only the internal binary header.
2	Check both.

Details

Polynomial used is:

x^32+x^26+x^23+x^22+x^16+x^12+x^11+x^10+x^8+x^7+x^5+x^4+x^2+x+1

See crc32.c in the public domain source code for zlib for more information.

See Also

StringCRC, **Hash**, **WaveHash**, **WaveModCount**

WaveDataToString

WaveDataToString(*wave*)

The WaveDataToString function returns a string containing an exact copy of the data from the numeric wave wave. The returned string may contain null bytes.

The WaveDataToString function was added in Igor Pro 9.00.

If *wave* has 0 points, WaveDataToStrings return an empty string. If *wave* is an invalid wave reference or a non-numeric wave, the WaveDataToStrings returns a null string.

If the size of the data in the wave is larger than about 2^31 bytes, the function returns a null string because Igor strings are limited to about 2GB in size. You can determine the size of the wave's data by multiplying the number of points by the size per point (double=8 bytes/point, float=4 bytes/point, etc.). The per-point size is doubled for complex waves.