

WaveTracking

WaveTracking [/FREE /LBL /LOCL] keyword

The WaveTracking operation is a debugging aid that can help you determine if your code creates waves and fails to kill them. This is especially helpful for finding free wave leaks. For background information, see **Wave Reference Counting** on page IV-205, **Free Wave Leaks** on page IV-94 and **Wave Tracking** on page IV-207.

WaveTracking was added in Igor Pro 9.00.

A fast and lightweight complement to wave tracking is **IgorInfo(16)**. For details, see **Detecting Wave Leaks** on page IV-206.

Flags

There are three flags that tell the WaveTracking operation what category of waves an invocation of the WaveTracking operation is aimed at. You can simultaneously track or count all three categories of waves, but you must use separate invocations of the operation to control or query the tracking of each category. See **Wave Tracking** on page IV-207 for a discussion of the wave categories.

You must include one of these flags.

- | | |
|-------|--|
| /FREE | Specifies that the current command applies to counting or tracking free waves. |
| /LBL | Specifies that the current command applies to counting or tracking global waves (waves in the main data hierarchy starting with the root data folder). |
| /LOCL | Specifies that the current command applies to counting or tracking local waves (waves contained by a free data folder). |
| /Q | Used with the dump keyword to suppress the information printed to the history. |

WaveTracking

Keywords

counter	Turns on wave tracking in <i>counter</i> mode for the category specified by /FREE, /GLBL, or /LOCL. Clears any existing count or tracking information for that specified category.
tracker	Turns on wave tracking in <i>tracker</i> mode for the category specified by /FREE, /GLBL, or /LOCL. In this mode, a list of waves of the specified category is kept. Clears any existing count or tracking information for that category.
count	Stores the count of waves in the variable V_numWaves for the category specified by /FREE, /GLBL, or /LOCL. The count is the number of waves created and not killed since <i>counter</i> or <i>tracker</i> mode was turned on for that category.
dump[=n]	Prints information into the history for the category specified by /FREE, /GLBL, or /LOCL. In <i>counter</i> mode, it prints just the number of waves. In <i>tracker</i> mode, it prints a line showing the count, and a number of lines showing the name of each wave that was created but not killed since tracking began, and the wave's reference count. If waves are global or local waves, the name of the containing data folder is also printed. If you omit <i>n</i> , the list is limited to 10 lines. Otherwise <i>n</i> sets the maximum number of lines to print. Due to the method used for tracking, the list is in random order. In addition, in <i>tracker</i> mode it creates a string variable S_waveTracker containing the same information. See below. Use the /Q flag to suppress the history printout.
status	Stores a number for the category specified by /FREE, /GLBL, or /LOCL indicating the type of tracking or zero into the variable V_Flag. 0 means no counting or tracking, 1 means counting, 2 means tracking.
stop	Stops wave tracking for the category specified by /FREE, /GLBL, or /LOCL and clears the count and list of waves of that category.

Details

In *counter* mode, each time a wave in the specified category is created the counter is incremented. Each time a wave is killed, the counter is decremented. If you start counting after some waves have been created, and get the count after killing those waves, it is possible for the count to become negative.

In *tracker* mode, the count is the number of waves created and not killed since you started tracking; it cannot be negative.

Creation of short waves takes approximately twice as long when *tracker* mode is turned on. Using *counter* mode has negligible effect on performance.

By default, all free waves have the name _free_, which limits the usefulness of the tracker dump. Starting with Igor Pro 9.00, to aid wave leak investigation, both **NewFreeWave** and **Make**/FREE have options for giving names to free waves. See **Free Wave Names** on page IV-95.

Output Variables

These variables are created and set by all keywords.

V_Flag	Indicates the type of tracking currently being used. The values are: 0: Not tracking or counting 1: Counter mode 2: Tracker mode
V_numWaves	The number of waves of the specified category created and not killed since the last time the counter or tracker keywords were used. If no tracking is enabled, this value will be zero. Set to zero if no tracking is currently enabled.
S_waveTracker	When you use the dump keyword in tracker mode, this string variable is created with a list of waves created since tracking started. The contents are a list of keyword-value strings separated by a carriage return. Each line of the contents is a keyword-value string containing the name, reference count and data folder for one of the list waves.

Examples

```

WaveTracking/GLBL counter          // start global tracker in counter mode
Make/O/N=1 jack, jill              // make two waves
WaveTracking/GLBL count            // ask for the count of waves in V_numWaves
print V_numWaves                  // print "2" in the history
WaveTracking/GLBL stop             // stops counting waves
KillWaves jack, jill               // so that we can count them all over again
WaveTracking/GLBL tracker          // start global tracker in tracker mode
Make/O/N=1 jack, jill              // make two waves
WaveTracking/GLBL count            // ask for the count of waves in V_numWaves
print V_numWaves                  // print "2" in the history
WaveTracking/GLBL dump             // ask for the history report on waves created
print S_waveTracker                // print the info string to the history
WaveTracking/GLBL stop             // stops counting waves

```

The dump keyword above prints this in the history:

Since tracking began, 2 global waves have been created and not killed.
 Wave 'jill'; data folder: 'root'; refcount: 1
 Wave 'jack'; data folder: 'root'; refcount: 1

The print command prints this:

```

WAVE:jill;REFCOUNT:1;DF:root;
WAVE:jack;REFCOUNT:1;DF:root;

```

To extract information from S_waveTracker:

```

print StringFromList(1, S_waveTracker, "\r")// extract and print second line
String str = StringFromList(0, S_waveTracker, "\r")// extract first line of information into string variable
print StringByKey("WAVE", str)// print the name of the wave from the first line
print StringByKey("DF", str)// print the wave's data folder from the first line

```

The first line prints the entire second line from S_waveTracking:

```

WAVE:jack;REFCOUNT:1;DF:root;

```

The third line extracts the wave name from the extracted second line and prints it to the history; the fourth line prints the data folder containing that wave:

```

jill
root

```

One more example showing both global and free trackers in use together:

```

WaveTracking/GLBL tracker
WaveTracking/FREEtracker
Make/N=3/WAVE wavewave          // make a global wave reference wave
// make three named free waves with references in the wave wave
wavewave = NewFreeWave(2, 1, "free_"+num2str(p))
WaveTracking/GLBL/Q dump         // /Q: we only want the S_waveTracker info
print S_waveTracker
WaveTracking/FREE/Q dump         // /Q: we only want the S_waveTracker info
print S_waveTracker
WaveTracking/GLBL stop            // stops counting waves
WaveTracking/FREE stop            // stops counting waves

```

The first print statement prints this:

```

WAVE:wavewave;REFCOUNT:1;DF:root;

```

The second print statement prints this, though the ordering of the lines will be different each time the code above is run again. These are free waves, which do not have a data folder:

```

WAVE:free_2;REFCOUNT:1;DF:;
WAVE:free_0;REFCOUNT:1;DF:;
WAVE:free_1;REFCOUNT:1;DF:;

```