See Also

GetRTLocInfo

GetRTLocInfo

GetRTLocInfo(*code*)

GetRTLocInfo is used for profiling Igor procedures.

You will typically not call GetRTLocInfo directly but instead will use it through FunctionProfiling.ipf which you can access using this include statement:

```
#include <FunctionProfiling>
```

GetRTLocation is called from an Igor preemptive thread to monitor the main thread. It returns a key/value string containing information about the procedure location associated with code or "" if the location could not be found.

Parameters

code is the result from a very recent call to **GetRTLocation**.

Details

The format of the result string is:

```
"PROCNAME:name;LINE:line;FUNCNAME:name;"
```

As of Igor Pro 7.03, if the code is in an independent module other than ProcGlobal then this appears at the beginning of the result string:

```
IMNAME:inName;
```

The line number is padded with zeros to facilitate sorting.

See Also

GetRTLocation

GetRTStackInfo

GetRTStackInfo(*selector*)

The GetRTStackInfo function returns information about "runtime stack" (the chain of macros and functions that are executing).

Details

If *selector* is 0, GetRTStackInfo returns a semicolon-separated list of the macros and procedures that are executing. This list is the same you would see in the debugger's stack list.

The currently executing macro or function is the last item in the list, the macro or function that started execution is the first item in the list.

If *selector* is 1, it returns the name of the currently executing function or macro.

If *selector* is 2, it returns the name of the calling function or macro.

If *selector* is 3, GetRTStackInfo returns a semicolon-separated list of routine names, procedure file names and line numbers. This is intended for advanced debugging by advanced programmers only.

For example, if RoutineA in procedure file ProcA.ipf calls RoutineB in procedure file ProcB.ipf, and RoutineB calls GetRTStackInfo(3), it will return:

```
RoutineA,ProcA.ipf,7;RoutineB,ProcB.ipf,12;
```

The numbers 7 and 12 would be the actual numbers of the lines that were executing in each routine. Line numbers are zero-based.

When called from a function started by **MultiThread** or **ThreadStart** the runtime stack information begins with the function that started threaded execution.

In future versions of Igor, *selector* may request other kinds of information.

Main Thread Example

```
Function Called()
   Print "Called by " + GetRTStackInfo(2) + "()"
```