The “MultiThread Mandelbrot Demo” experiment.

MultiThreadingControl

MultiThreadingControl *keyword* [=*value*]

The MultiThreadingControl operation allows you to control how automatic multithreading works with those IGOR operations that support it. Automatic multithreading is described below under Details.

For most purposes you will not need to use this operation.

The MultiThreadingControl operation was added in Igor Pro 7.00.

Keywords

getMode — Writes the current mode value into the variable V_autoMultiThread.

getThresholds — Creates the wave W_MultiThreadingArraySizes in the current data folder. See **Automatic Multithreading Thresholds** below for details.

setMode=*m* — Sets the mode for automatic multithreading. The mode controls the circumstances in which automatic multithreading is enabled.

m=0: Disables automatic multithreading unconditionally.

m=1: Enables automatic multithreading based on operation-specific thresholds for operations called from the main thread only. This is the default setting.

m=4: Enables automatic multithreading based on operation-specific thresholds for operations called from the main thread and from user-created explicit threads.

m=8: Enables automatic multithreading unconditionally - regardless of thresholds or the type of the calling thread.

You can not combine modes by ORing. The only valid values for *m* are those shown above.

setThresholds=*tWave* — Sets the thresholds for automatic multithreading. See **Automatic Multithreading Thresholds** below for details.

Details

Some IGOR operations and functions have internal code that can execute calculations in parallel using multiple threads. These operations are marked as "Automatically Multithreaded" in the Command Help pane of the Igor Help Browser.

Running on multiple threads reduces the time required for number-crunching tasks on multi-processor machines when the benefit of using multiple processors exceeds the overhead of running in multiple threads. This is usually the case only for large-scale jobs.

By default Igor uses automatic multithreading in operations that support it when the number of calculations exceeds a threshold value. This is called "automatic multithreading" to distinguish it from the explicit multithreading that you can instruct Igor to do. Explicit multithreading is described under **ThreadSafe Functions and Multitasking** on page IV-329. You don't need to do anything to benefit from automatic multithreading.

By default automatic multithreading is enabled for operations called from the main thread and disabled for operations called from explicit threads that you create (mode=1). You can change this using the setMode keyword described above.

The state of automatic multithreading is not saved with the experiment. It is initialized to mode=1 with default thresholds every time you start IGOR.

Automatic Multithreading Thresholds

Executing these commands

```
MultiThreadingControl getThresholds
Edit W_MultiThreadingArraySizes.ld
```