
GreenletProfiler Documentation

Release 0.1

A. Jesse Jiryu Davis

January 26, 2014

1	Contents	3
1.1	GreenletProfiler functions	3
1.2	Yappi Statistics Classes	3
	Python Module Index	5

A greenlet-aware performance profiler, suitable for analyzing Gevent applications or other Python programs that use [greenlets](#).

```
GreenletProfiler.set_clock_type('cpu')
GreenletProfiler.start()
my_function()
GreenletProfiler.stop()
stats = GreenletProfiler.get_func_stats()
stats.print_all()
stats.save('profile.callgrind', type='callgrind')
```

GreenletProfiler is based on [Yappi v0.82](#) and wraps its API.

1.1 GreenletProfiler functions

`GreenletProfiler.start(builtins=False, profile_threads=True)`

Starts profiling all threads and all greenlets.

This function can be called from any thread at any time. Resumes profiling if `stop()` was called previously.

- *builtins*: Profile builtin functions used by standart Python modules.
- *profile_threads*: Profile all threads if `True`, else profile only the calling thread.

`GreenletProfiler.stop()`

Stops the currently running yappi instance.

The same profiling session can be resumed later by calling `start()`.

`GreenletProfiler.clear_stats()`

Clears all of the profile results.

`GreenletProfiler.get_func_stats()`

Gets the function profiler results with given filters and returns an iterable.

`GreenletProfiler.get_thread_stats()`

Gets the thread profiler results with given filters and returns an iterable.

`GreenletProfiler.is_running()`

Returns true if the profiler is running, false otherwise.

`GreenletProfiler.get_clock_type()`

Returns the OS api used for timing plus the precision and the clock type information in a dict.

`GreenletProfiler.set_clock_type(type)`

Sets the internal clock type for timing. Profiler shall not have any previous stats. Otherwise an exception is thrown.

`GreenletProfiler.get_mem_usage()`

Returns the internal memory usage of the profiler itself.

`GreenletProfiler.convert2pstats(stats)`

Converts the internal stat type of yappi(which is returned by a call to `YFuncStats.get()`) as pstats object.

1.2 Yappi Statistics Classes

class `yappi.YFuncStat` (*values*)

Class holding information for function stats.

class `yappi.YChildFuncStat` (*values*)

Class holding information for children function stats.

class `yappi.YFuncStats` (*files*=[])

print_all (*out*=<open file '<stdout>', mode 'w' at 0x7fe91f121150>)

Prints all of the function profiler results to a given file. (stdout by default)

class `yappi.YThreadStat` (*values*)

Class holding information for thread stats.

class `yappi.YThreadStats`

print_all (*out*=<open file '<stdout>', mode 'w' at 0x7fe91f121150>)

Prints all of the thread profiler results to a given file. (stdout by default)

g

GreenletProfiler, 3