

Navigation

- [index](#)
- [modules](#) |
- [next](#) |
- [previous](#) |
- [Python v2.6.4 documentation](#) »
- [The Python Standard Library](#) »
- [27. Debugging and Profiling](#) »

27.5. hotshot — High performance logging profiler¶

New in version 2.2.

This module provides a nicer interface to the `_hotshot` C module. Hotshot is a replacement for the existing `profile` module. As it's written mostly in C, it should result in a much smaller performance impact than the existing `profile` module.

Note

The `hotshot` module focuses on minimizing the overhead while profiling, at the expense of long data post-processing times. For common usage it is recommended to use `cProfile` instead. `hotshot` is not maintained and might be removed from the standard library in the future.

Changed in version 2.5: The results should be more meaningful than in the past: the timing core contained a critical bug.

Note

The `hotshot` profiler does not yet work well with threads. It is useful to use an unthreaded script to run the profiler over the code you're interested in measuring if at all possible.

```
class hotshot.Profile(logfile[, lineevents[, linetimings]])¶
```

The profiler object. The argument `logfile` is the name of a log file to use for logged profile data. The argument `lineevents` specifies whether to generate events for every source line, or just on function call/return. It defaults to 0 (only log function call/return). The argument `linetimings` specifies whether to record timing information. It defaults to 1 (store timing information).

27.5.1. Profile Objects¶

Profile objects have the following methods:

```
Profile.addinfo(key, value)¶
```

Add an arbitrary labelled value to the profile output.

```
Profile.close()¶
```

Close the logfile and terminate the profiler.

```
Profile.fileno()¶
```

Return the file descriptor of the profiler's log file.

```
Profile.run(cmd)¶
```

Profile an `exec`-compatible string in the script environment. The globals from the `__main__` module are used as both the globals and locals for the script.

```
Profile.runcall(func, *args, **keywords)¶
```

Profile a single call of a callable. Additional positional and keyword arguments may be passed along; the result of the call is returned, and exceptions are allowed to propagate cleanly, while ensuring that profiling is disabled on the way out.

```
Profile.runtcx(cmd, globals, locals)¶
```

Evaluate an `exec`-compatible string in a specific environment. The string is compiled before profiling begins.

```
Profile.start()¶
```

Start the profiler.

```
Profile.stop()¶
```

Stop the profiler.

27.5.2. Using hotshot data¶

New in version 2.2.

This module loads hotshot profiling data into the standard `pstats` Stats objects.

```
hotshot.stats.load(filename)¶
```

Load hotshot data from `filename`. Returns an instance of the `pstats.Stats` class.

See also

27.5.3. Example Usage¶

Note that this example runs the Python “benchmark” pystones. It can take some time to run, and will produce large output files.

```
>>> import hotshot, hotshot.stats, test.pystone
>>> prof = hotshot.Profile("stones.prof")
>>> benchtime, stones = prof.runcall(test.pystone.pystones)
>>> prof.close()
>>> stats = hotshot.stats.load("stones.prof")
>>> stats.strip_dirs()
>>> stats.sort_stats('time', 'calls')
>>> stats.print_stats(20)
      850004 function calls in 10.090 CPU seconds

Ordered by: internal time, call count

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
      1    3.295    3.295    10.090    10.090  pystone.py:79(Proc0)
150000    1.315    0.000     1.315     0.000  pystone.py:203(Proc7)
 50000    1.313    0.000     1.463     0.000  pystone.py:229(Func2)
.
.
.
```

[Table Of Contents](#)

[27.5. hotshot — High performance logging profiler](#)

- [27.5.1. Profile Objects](#)
- [27.5.2. Using hotshot data](#)
- [27.5.3. Example Usage](#)

Previous topic

[27.4. The Python Profilers](#)

Next topic

[27.6. timeit — Measure execution time of small code snippets](#)

This Page

- [Show Source](#)

Navigation

- [index](#)
- [modules](#) |
- [next](#) |
- [previous](#) |
- [Python v2.6.4 documentation](#) »
- [The Python Standard Library](#) »
- [27. Debugging and Profiling](#) »

© [Copyright](#) 1990-2010, Python Software Foundation.

The Python Software Foundation is a non-profit corporation. [Please donate.](#)

Last updated on Feb 26, 2010. Created using [Sphinx](#) 0.6.3.