

:mod:`trace` --- Trace or track Python statement execution

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 1); [backlink](#)

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 4)

Unknown directive type "module".

```
.. module:: trace
   :synopsis: Trace or track Python statement execution.
```

Source code: :source:`Lib/trace.py`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 7); [backlink](#)

Unknown interpreted text role "source".

The `:mod:`trace`` module allows you to trace program execution, generate annotated statement coverage listings, print caller/callee relationships and list functions executed during a program run. It can be used in another program or from the command line.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 11); [backlink](#)

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 16)

Unknown directive type "seealso".

```
.. seealso::

   `Coverage.py <https://coverage.readthedocs.io/>`_
   A popular third-party coverage tool that provides HTML
   output along with advanced features such as branch coverage.
```

Command-Line Usage

The `:mod:`trace`` module can be invoked from the command line. It can be as simple as

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 27); [backlink](#)

Unknown interpreted text role "mod".

```
python -m trace --count -C . somefile.py ...
```

The above will execute `:file:`somefile.py`` and generate annotated listings of all Python modules imported during the execution into the current directory.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 32); [backlink](#)

Unknown interpreted text role "file".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-

main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 35)

Unknown directive type "program".

```
.. program:: trace
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 37)

Unknown directive type "cmdoption".

```
.. cmdoption:: --help
```

Display usage and exit.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 41)

Unknown directive type "cmdoption".

```
.. cmdoption:: --version
```

Display the version of the module and exit.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 45)

Unknown directive type "versionadded".

```
.. versionadded:: 3.8
```

Added ``--module`` option that allows to run an executable module.

Main options

At least one of the following options must be specified when invoking `:mod:trace`. The `:option:--listfuncs <-P>` option is mutually exclusive with the `:option:--trace <-t>` and `:option:--count <-c>` options. When `:option:--listfuncs <-P>` is provided, neither `:option:--count <-c>` nor `:option:--trace <-t>` are accepted, and vice versa.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 51); [backlink](#)

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 51); [backlink](#)

Unknown interpreted text role "option".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 51); [backlink](#)

Unknown interpreted text role "option".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 51); [backlink](#)

Unknown interpreted text role "option".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 51); [backlink](#)

Unknown interpreted text role "option".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 51); [backlink](#)

Unknown interpreted text role "option".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 51); [backlink](#)

Unknown interpreted text role "option".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 57)

Unknown directive type "program".

```
.. program:: trace
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 59)

Unknown directive type "cmdoption".

```
.. cmdoption:: -c, --count
```

Produce a set of annotated listing files upon program completion that shows how many times each statement was executed. See also
:option:``--coverdir <-C>`, :option:``--file <-f>` and
:option:``--no-report <-R>` below.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 66)

Unknown directive type "cmdoption".

```
.. cmdoption:: -t, --trace
```

Display lines as they are executed.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 70)

Unknown directive type "cmdoption".

```
.. cmdoption:: -l, --listfuncs
```

Display the functions executed by running the program.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 74)

Unknown directive type "cmdoption".

```
.. cmdoption:: -r, --report
```

Produce an annotated list from an earlier program run that used the
:option:``--count <-c>` and :option:``--file <-f>` option. This does not
execute any code.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 80)

Unknown directive type "cmdoption".

```
.. cmdoption:: -T, --trackcalls
```

Display the calling relationships exposed by running the program.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 87)

Unknown directive type "program".

```
.. program:: trace
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 89)

Unknown directive type "cmdoption".

```
.. cmdoption:: -f, --file=<file>
```

Name of a file to accumulate counts over several tracing runs. Should be used with the :option:``--count <-c>` option.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 94)

Unknown directive type "cmdoption".

```
.. cmdoption:: -C, --coverdir=<dir>
```

Directory where the report files go. The coverage report for ``package.module`` is written to file :file:``{dir}/{package}/{module}.cover``.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 99)

Unknown directive type "cmdoption".

```
.. cmdoption:: -m, --missing
```

When generating annotated listings, mark lines which were not executed with ``>>>>>>``.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 104)

Unknown directive type "cmdoption".

```
.. cmdoption:: -s, --summary
```

When using :option:``--count <-c>` or :option:``--report <-r>``, write a brief summary to stdout for each file processed.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 109)

Unknown directive type "cmdoption".

```
.. cmdoption:: -R, --no-report
```

Do not generate annotated listings. This is useful if you intend to make several runs with :option:``--count <-c>``, and then produce a single set of annotated listings at the end.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 115)

Unknown directive type "cmdoption".

```
.. cmdoption:: -g, --timing
```

Prefix each line with the time since the program started. Only used while tracing.

Filters

These options may be repeated multiple times.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 125)

Unknown directive type "program".

```
.. program:: trace
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 127)

Unknown directive type "cmdoption".

```
.. cmdoption:: --ignore-module=<mod>
```

Ignore each of the given module names and its submodules (if it is a package). The argument can be a list of names separated by a comma.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 132)

Unknown directive type "cmdoption".

```
.. cmdoption:: --ignore-dir=<dir>
```

Ignore all modules and packages in the named directory and subdirectories. The argument can be a list of directories separated by :data:`os.pathsep`.

Programmatic Interface

Create an object to trace execution of a single statement or expression. All parameters are optional. *count* enables counting of line numbers. *trace* enables line execution tracing. *countfuncs* enables listing of the functions called during the run. *countcallers* enables call relationship tracking. *ignoremods* is a list of modules or packages to ignore. *ignoredirs* is a list of directories whose modules or packages should be ignored. *infile* is the name of the file from which to read stored count information. *outfile* is the name of the file in which to write updated count information. *timing* enables a timestamp relative to when tracing was started to be displayed.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 156)

Unknown directive type "method".

```
.. method:: run(cmd)
```

Execute the command and gather statistics from the execution with the current tracing parameters. *cmd* must be a string or code object, suitable for passing into :func:`exec`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 162)

Unknown directive type "method".

```
.. method:: runctx(cmd, globals=None, locals=None)
```

Execute the command and gather statistics from the execution with the current tracing parameters, in the defined global and local environments. If not defined, *globals* and *locals* default to empty dictionaries.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 169)

Unknown directive type "method".

```
.. method:: runfunc(func, /, *args, **kws)
```

Call `*func*` with the given arguments under control of the `:class:`Trace`` object with the current tracing parameters.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 174)

Unknown directive type "method".

```
.. method:: results()
```

Return a `:class:`CoverageResults`` object that contains the cumulative results of all previous calls to ```run```, ```runctx``` and ```runfunc``` for the given `:class:`Trace`` instance. Does not reset the accumulated trace results.

A container for coverage results, created by `meth:`Trace.results``. Should not be created directly by the user.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 183); [backlink](#)

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 186)

Unknown directive type "method".

```
.. method:: update(other)
```

Merge in data from another `:class:`CoverageResults`` object.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library) trace.rst, line 190)

Unknown directive type "method".

```
.. method:: write_results(show_missing=True, summary=False, coverdir=None)
```

Write coverage results. Set `*show_missing*` to show lines that had no hits. Set `*summary*` to include in the output the coverage summary per module. `*coverdir*` specifies the directory into which the coverage result files will be output. If ```None```, the results for each source file are placed in its directory.

A simple example demonstrating the use of the programmatic interface:

```
import sys
import trace

# create a Trace object, telling it what to ignore, and whether to
# do tracing or line-counting or both.
tracer = trace.Trace(
    ignoredirs=[sys.prefix, sys.exec_prefix],
    trace=0,
    count=1)

# run the new command using the given tracer
tracer.run('main()')

# make a report, placing output in the current directory
r = tracer.results()
r.write_results(show_missing=True, coverdir=".")
```