:mod:`faulthandler` --- Dump the Python traceback

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

Unknown directive type "module".

```
.. module:: faulthandler
   :synopsis: Dump the Python traceback.
```

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

Unknown directive type "versionadded".

```
.. versionadded:: 3.3
```

This module contains functions to dump Python tracebacks explicitly, on a fault, after a timeout, or on a user signal. Call :func:`faulthandler.enable` to install fault handlers for the :const:`SIGSEGV`, :const:`SIGFPE`, :const:`SIGABRT`, :const:`SIGBUS`, and :const:`SIGILL` signals. You can also enable them at startup by setting the :envvar:`PYTHONFAULTHANDLER` environment variable or by using the :option:`-X` `faulthandler` command line option.

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

Unknown interpreted text role "func".

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

Unknown interpreted text role "const".

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

Unknown interpreted text role "const".

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

Unknown interpreted text role "const".

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

Unknown interpreted text role "const".

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

Unknown interpreted text role "const".

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

Unknown interpreted text role "envvar".

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

Unknown interpreted text role "option".

The fault handler is compatible with system fault handlers like Apport or the Windows fault handler. The module uses an alternative stack for signal handlers if the :c:func:\`sigaltstack\` function is available. This allows it to dump the traceback even on a stack overflow.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]faulthandler.rst`, line 18); *backlink*

Unknown interpreted text role "c:func".

The fault handler is called on catastrophic cases and therefore can only use signal-safe functions (e.g. it cannot allocate memory on the heap). Because of this limitation traceback dumping is minimal compared to normal Python tracebacks:

- Only ASCII is supported. The `backslashreplace` error handler is used on encoding.
- Each string is limited to 500 characters.
- Only the filename, the function name and the line number are displayed. (no source code)
- It is limited to 100 frames and 100 threads.
- The order is reversed: the most recent call is shown first.

By default, the Python traceback is written to :data:\`sys.stderr\`. To see tracebacks, applications must be run in the terminal. A log file can alternatively be passed to :func:\`faulthandler.enable\`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]faulthandler.rst`, line 36); *backlink*

Unknown interpreted text role "data".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]faulthandler.rst`, line 36); *backlink*

Unknown interpreted text role "func".

The module is implemented in C, so tracebacks can be dumped on a crash or when Python is deadlocked.

The :ref:\`Python Development Mode <devmode>\` calls :func:\`faulthandler.enable\` at Python startup.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]faulthandler.rst`, line 43); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]faulthandler.rst`, line 43); *backlink*

Unknown interpreted text role "func".

Dumping the traceback

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]faulthandler.rst`, line 50)

Unknown directive type "function".

```
.. function:: dump_traceback(file=sys.stderr, all_threads=True)

   Dump the tracebacks of all threads into *file*. If *all_threads* is
   ``False``, dump only the current thread.

   .. versionchanged:: 3.5
      Added support for passing file descriptor to this function.
```

Fault handler state

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]faulthandler.rst, line 62)

Unknown directive type "function".

```
.. function:: enable(file=sys.stderr, all_threads=True)

   Enable the fault handler: install handlers for the :const:`SIGSEGV`,
   :const:`SIGFPE`, :const:`SIGABRT`, :const:`SIGBUS` and :const:`SIGILL`
   signals to dump the Python traceback. If *all_threads* is ``True``,
   produce tracebacks for every running thread. Otherwise, dump only the current
   thread.

   The *file* must be kept open until the fault handler is disabled: see
   :ref:`issue with file descriptors <faulthandler-fd>`.

   .. versionchanged:: 3.5
      Added support for passing file descriptor to this function.

   .. versionchanged:: 3.6
      On Windows, a handler for Windows exception is also installed.

   .. versionchanged:: 3.10
      The dump now mentions if a garbage collector collection is running
      if *all_threads* is true.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]faulthandler.rst, line 83)

Unknown directive type "function".

```
.. function:: disable()

   Disable the fault handler: uninstall the signal handlers installed by
   :func:`enable`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]faulthandler.rst, line 88)

Unknown directive type "function".

```
.. function:: is_enabled()

   Check if the fault handler is enabled.
```

Dumping the tracebacks after a timeout

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]faulthandler.rst, line 96)

Unknown directive type "function".

```
.. function:: dump_traceback_later(timeout, repeat=False, file=sys.stderr, exit=False)

   Dump the tracebacks of all threads, after a timeout of *timeout* seconds, or
   every *timeout* seconds if *repeat* is ``True``.  If *exit* is ``True``, call
   :c:func:`_exit` with status=1 after dumping the tracebacks.  (Note
   :c:func:`_exit` exits the process immediately, which means it doesn't do any
   cleanup like flushing file buffers.) If the function is called twice, the new
   call replaces previous parameters and resets the timeout. The timer has a
   sub-second resolution.

   The *file* must be kept open until the traceback is dumped or
   :func:`cancel_dump_traceback_later` is called: see :ref:`issue with file
   descriptors <faulthandler-fd>`.

   This function is implemented using a watchdog thread.

   .. versionchanged:: 3.7
      This function is now always available.

   .. versionchanged:: 3.5
      Added support for passing file descriptor to this function.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]faulthandler.rst, line 118)

Unknown directive type "function".

```
.. function:: cancel_dump_traceback_later()

   Cancel the last call to :func:`dump_traceback_later`.
```

Dumping the traceback on a user signal

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]faulthandler.rst, line 126)

Unknown directive type "function".

```
.. function:: register(signum, file=sys.stderr, all_threads=True, chain=False)

   Register a user signal: install a handler for the *signum* signal to dump
   the traceback of all threads, or of the current thread if *all_threads* is
   ``False``, into *file*. Call the previous handler if chain is ``True``.

   The *file* must be kept open until the signal is unregistered by
   :func:`unregister`: see :ref:`issue with file descriptors <faulthandler-fd>`.

   Not available on Windows.

   .. versionchanged:: 3.5
      Added support for passing file descriptor to this function.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]faulthandler.rst, line 140)

Unknown directive type "function".

```
.. function:: unregister(signum)

   Unregister a user signal: uninstall the handler of the *signum* signal
   installed by :func:`register`. Return ``True`` if the signal was registered,
   ``False`` otherwise.

   Not available on Windows.
```

Issue with file descriptors

:func:`enable`, :func:`dump_traceback_later` and :func:`register` keep the file descriptor of their *file* argument. If the file is closed and its file descriptor is reused by a new file, or if :func:`os.dup2` is used to replace the file descriptor, the traceback will be written into a different file. Call these functions again each time that the file is replaced.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]faulthandler.rst, line 154); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]faulthandler.rst, line 154); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]faulthandler.rst, line 154); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]faulthandler.rst`, line 154); *backlink*

Unknown interpreted text role "func".

Example

Example of a segmentation fault on Linux with and without enabling the fault handler:

```
$ python3 -c "import ctypes; ctypes.string_at(0)"
Segmentation fault

$ python3 -q -X faulthandler
>>> import ctypes
>>> ctypes.string_at(0)
Fatal Python error: Segmentation fault

Current thread 0x00007fb899f39700 (most recent call first):
  File "/home/python/cpython/Lib/ctypes/__init__.py", line 486 in string_at
  File "<stdin>", line 1 in <module>
Segmentation fault
```