

Removed a too-strict assertion that failed for certain f-strings, such as `eval("\n")` and `eval("\r")`.

Move `co_extra_freelfuncs` to not be per-thread to avoid crashes

Fixed parsing backslashes in f-strings.

Fixed various segfaults with dict when input collections are mutated during searching, inserting or comparing. Based on patches by Duane Griffin and Tim Mitchell.

Fixed type `__setattr__()` and type `__delattr__()` for non-interned attribute names. Based on patch by Eryk Sun.

If a `KeyboardInterrupt` happens when the interpreter is in the middle of resuming a chain of nested 'yield from' or 'await' calls, it's now correctly delivered to the innermost frame.

`sys.getsizeof()` on a code object now returns the sizes which includes the code struct and sizes of objects which it references. Patch by Dong-hee Na.

Fix memory usage regression of set and frozenset object.

Fixed error messages in the `index()` method of tuple, list and deque when pass indices of wrong type.

Show correct error messages when any of the `pthread_*` calls in `thread_pthread.h` fails.

`bool(range)` works even if `len(range)` raises `:exc:'OverflowError'`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ (cpython-main) (Misc) (NEWS.d) 3.6.2rc1.rst, line 106); [backlink](#)

Unknown interpreted text role "exc".

Fix wrapping coroutine return values in `StopIteration`.

Fix an oversight that `%b` format for bytes should support objects follow the buffer protocol.

Fix a regression that bytes format may fail when containing zero bytes inside.

If `max_line_length=None` is specified while using the `Compat32` policy, it is no longer ignored. Patch by Mircea Cosbuc.

Functional API of enum allows to create empty enums. Patched by Dong-hee Na

Fix race condition between signal delivery and wakeup file descriptor. Patch by Nathaniel Smith.

`lib2to3` now recognizes `rb'...'` and `f'...'` strings.

`unittest.TestCase.assertRaises()` now manually breaks a reference cycle to not keep objects alive longer than expected.

`inspect.signature()` now supports callables with variable-argument parameters wrapped with `partialmethod`. Patch by Dong-hee Na.

Fix path calculation in `imp.load_package()`, fixing it for cases when a package is only shipped with bytecodes. Patch by Alexandru Ardelean.

Fixed comparison check for `ipaddress.ip_interface` objects. Patch by Sanjay Sundaresan.

`re.compile()` no longer raises a `BytesWarning` when compiling a bytes instance with misplaced inline modifier. Patch by Roy Williams.

Update expat copy from 2.1.1 to 2.2.0 to get fixes of CVE-2016-0718 and CVE-2016-4472. See <https://sourceforge.net/p/expat/bugs/537/> for more information.

Avoid race condition in multiprocessing cleanup (#2159)

The traceback no longer displayed for `SystemExit` raised in a callback registered by `atexit`.

Don't log exceptions if `Task/Future.cancel()` method was called.

Updates to typing module: Add generic `AsyncContextManager`, add support for `ContextManager` on all versions. Original PRs by Jelle Zijlstra and Ivan Levkivskyi

Fix ssl sockets leaks when connection is aborted in `asyncio/ssl` implementation. Patch by Michał Sghaier.

Closing transport during handshake process leaks open socket. Patch by Nikolay Kim

Fix waiter cancellation in `asyncio.Lock`. Patch by Mathieu Sornay.

On Windows, `subprocess.Popen.communicate()` now also ignore `EINVAL` on `stdin.write()` if the child process is still running but closed the pipe.

`inspect.isabstract()` now works during `__init_subclass__`. Patch by Nate Soares.

`ABCMeta.__new__` now accepts `**kwargs`, allowing abstract base classes to use keyword parameters in `__init_subclass__`. Patch by Nate Soares.

`faulthandler` now correctly filters and displays exception codes on Windows

Fix the problem that `logging.handlers.SysLogHandler` cannot handle IPv6 addresses.

Preserve generator state when `_random.Random.setstate()` raises an exception. Patch by Bryan Olson.

multiprocessing.Queue._feed background running thread do not break from main loop on exception.

Fix handling escape characters in HZ codec. Based on patch by Ma Lin.

Fix AttributeError when using SimpleQueue.empty() under *spawn* and *forkserver* start methods.

imaplib and poplib now catch the Windows socket WSAEINVAL error (code 10022) on shutdown(SHUT_RDWR): An invalid operation was attempted. This error occurs sometimes on SSL connections.

Warnings emitted when compile a regular expression now always point to the line in the user code. Previously they could point into inners of the re module if emitted from inside of groups or conditionals.

Fixed Task.cancel() can be ignored when the task is running coroutine and the coroutine returned without any more *await*.

contextlib.AbstractContextManager now supports anti-registration by setting `__enter__ = None` or `__exit__ = None`, following the pattern introduced in bpo-25958. Patch by Jelle Zijlstra.

Weaken the condition of deprecation warnings for inline modifiers. Now allowed several subsequential inline modifiers at the start of the pattern (e.g. '(?i) (?s) ...'). In verbose mode whitespaces and comments now are allowed before and between inline modifiers (e.g. '(?x) (?i) (?s) ...').

Fix range checking in GB18030 decoder. Original patch by Ma Lin.

Change resulted because of zipfile breakage. (See also: bpo-29094)

Removed the `__init__` methods of `_json`'s scanner and encoder. Misusing them could cause memory leaks or crashes. Now scanner and encoder objects are completely initialized in the `__new__` methods.

Avoid KeyboardInterrupt tracebacks in forkserver helper process when Ctrl-C is received.

Various updates to typing module: add typing.NoReturn type, use WrapperDescriptorType, minor bug-fixes. Original PRs by Jim Fasarakis-Hilliard and Ivan Levkivskyi.

Fix getsockname() for unbound AF_UNIX sockets on Linux.

Fixed leaks and crashes in errors handling in the parser module.

Fixed crashes in IOBase methods `__next__()` and `readlines()` when `readline()` or `__next__()` respectively return non-sizeable object. Fixed possible other errors caused by not checking results of `PyObject_Size()`, `PySequence_Size()`, or `PyMapping_Size()`.

Allowed calling the `close()` method of the zip entry writer object multiple times. Writing to a closed writer now always produces a `ValueError`.

`_io._IOBase.readlines` will check if it's closed first when `hint` is present.

Fixed race condition in `pathlib.mkdir` with `flags.parents=True`. Patch by Armin Rigo.

Fixed arbitrary unchaining of `RuntimeError` exceptions in `contextlib.contextmanager`. Patch by Siddharth Velankar.

Pickling and copying `ImportError` now preserves `name` and `path` attributes.

Fixed memory leaks in the `replace()` method of `datetime` and `time` objects when pass out of bound `fold` argument.

Fix a crash in `itertools.chain.from_iterable` when encountering long runs of empty iterables.

Fixed multiple crashes in `ElementTree` caused by race conditions and wrong types.

Fixed a bug in pools in `multiprocessing.pool` that raising an exception at the very first of an iterable may swallow the exception or make the program hang. Patch by Davin Potts and Xiang Zhang.

Avoid incorrect errors raised by `Path.mkdir(exist_ok=True)` when the OS gives priority to errors such as `EACCES` over `EEXIST`.

Release references to tasks, their arguments and their results as soon as they are finished in `multiprocessing.Pool`.

faulthandler: Restore the old `sigaltstack` during teardown. Patch by Christophe Zeitouny.

Fixed crashes in `repr` of recursive buffered file-like objects.

Fix crashes in `partial.__repr__` if the keys of `partial.keywords` are not strings. Patch by Michael Seifert.

`get_extra_info()` raises exception if `get` called on closed ssl transport. Patch by Nikolay Kim

Fixed possible failing or crashing `input()` if attributes "encoding" or "errors" of `sys.stdin` or `sys.stdout` are not set or are not strings.

Fix a bug that prevented array 'Q', 'L' and 'I' from accepting big intables (objects that have `__int__`) as elements. Patch by Oren Milman.

The `zipfile` module now accepts path-like objects for external paths.

`index()` and `count()` methods of `collections.abc.Sequence` now check identity before checking equality when do comparisons.

`SimpleXMLRPCDispatcher` no longer chains `KeyError` (or any other exception) to exception(s) raised in the dispatched methods. Patch by Petr Motejlek.

`path.resolve(strict=False)` no longer cuts the path after the first element not present in the filesystem. Patch by Antoine Pietri.

Fix several problems with IDLE's autocompletion box. The following should now work: clicking on selection box items; using the

scrollbar; selecting an item by hitting Return. Hangs on MacOSX should no longer happen. Patch by Louie Lu.

Add doc subsection about IDLE failure to start. Popup no-connection message directs users to this section.

Fix reference leaks in IDLE tests. Patches by Louie Lu and Terry Jan Reedy.

Add docstrings for textview.py and use PEP8 names. Patches by Cheryl Sabella and Terry Jan Reedy.

Help-about: use pep8 names and add tests. Increase coverage to 100%. Patches by Louie Lu, Cheryl Sabella, and Terry Jan Reedy.

Add `_utest` option to textview; add new tests. Increase coverage to 100%. Patches by Louie Lu and Terry Jan Reedy.

Function `PySlice_GetIndicesEx()` no longer replaced with a macro if `Py_LIMITED_API` is not set.

Add `--with-assertions` configure flag to explicitly enable `C assert()` checks. Defaults to off. `--with-pydebug` implies `--with-assertions`.

Fix out-of-tree builds of Python when configured with `--with--dtrace`.

Prevent unnecessary rebuilding of Python during `make test`, `make install` and some other make targets when configured with `--enable-optimizations`.

Don't regenerate generated files based on file modification time anymore: the action is now explicit. Replace `make touch` with `make regen-all`.

Fix `--enable-optimization` didn't work.

Add missing attribute related constants in curses documentation.

the link targets for `:func:'bytes'` and `:func:'bytearray'` are now their respective type definitions, rather than the corresponding builtin function entries. Use `:ref:'bytes' <func-bytes>` and `:ref:'bytearray' <func-bytearray>` to reference the latter. In order to ensure this and future cross-reference updates are applied automatically, the daily documentation builds now disable the default output caching features in Sphinx.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ (cpython-main) (Misc) (NEWS.d) 3.6.2rc1.rst, line 871); [backlink](#)

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ (cpython-main) (Misc) (NEWS.d) 3.6.2rc1.rst, line 871); [backlink](#)

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ (cpython-main) (Misc) (NEWS.d) 3.6.2rc1.rst, line 871); [backlink](#)

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ (cpython-main) (Misc) (NEWS.d) 3.6.2rc1.rst, line 871); [backlink](#)

Unknown interpreted text role "ref".

Add missing info of code object in inspect documentation.

`python-gdb.py` now supports also `method-wrapper (wrapperobject)` objects.

`test_thread`: `setUp()` now uses `support.threading_setup()` and `support.threading_cleanup()` to wait until threads complete to avoid random side effects on following tests. Initial patch written by Grzegorz Grzywacz.

Enhanced functions `swap_attr()` and `swap_item()` in the `test.support` module. They now work when delete replaced attribute or item inside the `with` statement. The old value of the attribute or item (or `None` if it doesn't exist) now will be assigned to the target of the `"as"` clause, if there is one.

Locate `msbuild.exe` on Windows when building rather than `vcvarsall.bat`

The build process on Windows no longer depends on Subversion, instead pulling external code from GitHub via a Python script. If Python 3.6 is not found on the system (via `py -3.6`), NuGet is used to download a copy of 32-bit Python.