

bytearray formatting no longer truncates on first null byte.

Do not include `<sys/random.h>` if it does not exist.

Fix a memory leak in split-table dictionaries: `setattr()` must not convert combined table into split table.

Correct the positioning of the syntax error caret for indented blocks. Based on patch by Michael Layzell.

Fixed bytes formatting of octals with zero padding in alternate form.

Fixed setting the offset attribute of `SyntaxError` by `PyErr_SyntaxLocationEx()` and `PyErr_SyntaxLocationObject()`.

`functools.lru_cache()` was susceptible to an obscure reentrancy bug caused by a monkey-patched `len()` function.

Fixed crash in `Py_DecodeLocale()` in debug build on Mac OS X when decode astral characters. Patch by Xiang Zhang.

Extra slash no longer added to `sys.path` components in case of empty compile-time `PYTHONPATH` components.

Fixed potential crash in `PyUnicode_AsDecodedObject()` in debug build.

Fixed possible memory leak in `_PyTraceback_Add()` and exception loss in `PyTraceBack_Here()`.

Added sanity checks and tests for `PyUnicode_CopyCharacters()`. Patch by Xiang Zhang.

The type of long range iterator is now registered as `Iterator`. Patch by Oren Milman.

The constructor of `range_iterator` now checks that `step` is not 0. Patch by Oren Milman.

Resolving special methods of uninitialized type now causes implicit initialization of the type instead of a fail.

`PyType_Ready()` now checks that `tp_name` is not `NULL`. Original patch by Niklas Koep.

Fixed possible crash when AST is changed in process of compiling it.

String constants with null character no longer interned.

Fix crash when GC runs during weakref callbacks.

String constants now interned recursively in tuples and frozensets.

Fixed misleading error message when `ImportError` called with invalid keyword args.

Fix incorrect type in error message from `complex(1.0, {2:3})`. Patch by Sounya Sharma.

Fallback on reading `/dev/urandom` device when the `getrandom()` syscall fails with `EPERM`, for example when blocked by `SECCOMP`.

Fix a regression in `zipimport`'s `compile_source()`. `zipimport` should use the same optimization level as the interpreter.

Fix corrupted result from `PyLong_FromLong(0)` when Python is compiled with `NSMALLPOSINTS = 0`.

Prevents `zipimport` from unnecessarily encoding a filename (patch by Eryk Sun)

`dictitems_contains` no longer swallows compare errors. (Patch by Xiang Zhang)

Properly clear out a generator's frame's backreference to the generator to prevent crashes in `frame.clear()`.

Fix a crash when a coroutine that has not been awaited is finalized with `warnings-as-errors` enabled.

Fix another issue found by PVS-Studio: Null pointer check after use of `'def'` in `_PyState_AddModule()`. Initial patch by Christian Heimes.

set literal evaluation order did not match documented behaviour.

Multi-phase extension module `import` now correctly allows the `m_methods` field to be used to add module level functions to instances of non-module types returned from `Py_create_mod`. Patch by Xiang Zhang.

The `round()` function accepted a second `None` argument for some types but not for others. Fixed the inconsistency by accepting `None` for all numeric types.

Warn if a submodule argument to `"python -m"` or `runpy.run_module()` is found in `sys.modules` after parent packages are imported, but before the submodule is executed.

Fix a `SystemError` in the implementation of `"raise"` statement. In a brand new thread, raise a `RuntimeError` since there is no active exception to reraise. Patch written by Xiang Zhang.

Standard `__import__()` no longer look up `"__import__"` in globals or builtins for importing submodules or `"from import"`. Fixed handling an error of non-string package name.

Respect the `PYTHONCASEOK` environment variable under Windows.

Make having too many statically nested blocks a `SyntaxError` instead of `SystemError`.

Fixed possible integer overflow in bytes and bytearray concatenations. Patch by Xiang Zhang.

Add integer overflow check in `bytearray.extend()`. Patch by Xiang Zhang.

Don't rely on wrapping for overflow check in `PySequence_Tuple()`. Patch by Xiang Zhang.

`__length_hint__()` of bytearray iterators no longer return a negative integer for a resized bytearray.

Fix memory leak in codeobject.c

inspect.getframeinfo() now correctly shows the first line of a context. Patch by Sam Breese.

Offsets in a ZIP file created with extern file object and modes "w" and "x" now are relative to the start of the file.

Fixed recursion errors in large or resized curses.textpad.Textbox. Based on patch by Tycho Andersen.

Fix weakrefs in the pure python version of collections.OrderedDict move_to_end() method. Contributed by Andra Bogildea.

curses.ascii predicates now work correctly with negative integers.

old keys should not remove new values from WeakValueDictionary when collecting from another thread.

Remove editor artifacts from Tix.py.

Fixed a crash when deallocate deep ElementTree.

Fix bugs in WeakValueDictionary.setdefault() and WeakValueDictionary.pop() when a GC collection happens in another thread.

Fixed a crash in resource.prlimit() when pass a sequence that doesn't own its elements as limits.

multiprocessing.set_forkserver_preload() would crash the forkserver process if a preloaded module instantiated some multiprocessing objects such as locks.

dbm.dumb now supports reading read-only files and no longer writes the index file when it is not changed.

In ctypes, prevent a crash calling the from_buffer() and from_buffer_copy() methods on abstract classes like Array.

Fix crash in os.spawnv() with no elements in args

Always raise ValueError for negative compileall.compile_dir(workers=...) parameter, even when multithreading is unavailable.

Fixed possible crash in _io.TextIOWrapper deallocator when the garbage collector is invoked in other thread. Based on patch by Sebastian Cufre.

LZMA compressor and decompressor no longer raise exceptions if given empty data twice. Patch by Benjamin Fogle.

Fixed segfault in curses's addch() with ncurses6.

tarfile.open() with mode "r" or "r:" now tries to open a tar file with compression before trying to open it without compression. Otherwise it had 50% chance failed with ignore_zeros=True.

The webbrowser module now supports Firefox 36+ and derived browsers. Based on patch by Oleg Broytman.

Fixed bugs in tkinter.ttk.LabeledScale and tkinter.Scale caused by representing the scale as float value internally in Tk. tkinter.IntVar now works if float value is set to underlying Tk variable.

calendar.TextCalendar().prmonth() no longer prints a space at the start of new line after printing a month's calendar. Patch by Xiang Zhang.

The textwrap.TextWrapper class now honors non-breaking spaces. Based on patch by Kaarle Ritvanen.

os.fwalk() no longer fails on broken links.

Fixed HList.header_exists() in tkinter.tix module by addin a workaround to Tix library bug.

shutil.make_archive() no longer add entry "/" to ZIP archive.

Make webbrowser support Chrome on Mac OS X.

Fix references leaked by pdb in the handling of SIGINT handlers.

Fixed writing ZIP files that starts not from the start of the file. Offsets in ZIP file now are relative to the start of the archive in conforming to the specification.

Fixed writing non-BMP characters with binary format in plistlib.

Fixed possible crashes when unpickle itertools objects from incorrect pickle data. Based on patch by John Leitch.

Fix possible integer overflows and crashes in the mmap module with unusual usage patterns.

Fix the ability to pass the --link-objects option to the distutils build_ext command.

Fixed calendar functions for extreme months: 0001-01 and 9999-12. Methods itermonthdays() and itermonthdays2() are reimplemented so that they don't call itermonthdates() which can cause datetime.date under/overflow.

Fixed possible use after free in the decompress() methods of the LZMADecompressor and BZ2Decompressor classes. Original patch by John Leitch.

Fixed possible crash in sqlite3.Connection.create_collation() if pass invalid string-like object as a name. Patch by Xiang Zhang.

Fix invalid exception handling in Lib/ctypes/macholib/dyld.py. Patch by Madison May.

Fixed support of default root window in the tkinter.tix module.

In the traceback module, restore the formatting of exception messages like "Exception: None". This fixes a regression introduced in 3.5a2.

Allow falsy values to be used for msg parameter of subTest().

Prevent memory leak in win32_ver().

Fix UnboundLocalError in socket._sendfile_use_sendfile.

Check for ERROR_ACCESS_DENIED in Windows implementation of os.stat(). Patch by Eryk Sun.

Prevent codecs.escape_encode() from raising SystemError when an empty bytestring is passed.

Get antigravity over HTTPS. Patch by Kaartic Sivaraam.

Enable WebSocket URL schemes in urllib.parse.urljoin. Patch by Gergely Imreh and Markus Holtermann.

Fixed buffer overrun in binascii.b2a_qp() and binascii.a2b_qp().

email.generator now replaces only \r and/or \n line endings, per the RFC, instead of all unicode line endings.

itertools.count() no longer rounds non-integer step in range between 1.0 and 2.0 to 1.

Update the lib2to3 grammar to handle the unpacking generalizations added in 3.5.

mailcap now respects the order of the lines in the mailcap files ("first match"), as required by RFC 1542. Patch by Michael Lazar.

Validates persist parameter when opening MSI database

xml.etree.ElementTree now preserves whitespaces in attributes (Patch by Duane Griffin. Reviewed and approved by Stefan Behnel.)

Fixed calculation of line length used for the base64 CTE in the new email policies.

Don't pass str(_charset) to MIMEText.set_payload(). Patch by Claude Paroz.

urllib now includes an Accept: */* header among the default headers. This makes the results of REST API requests more consistent and predictable especially when proxy servers are involved.

lib2to3.pgen3.driver.load_grammar() now creates a stable cache file between runs given the same Grammar.txt input regardless of the hash randomization setting.

Avoid zero-length memcpy() etc calls with null source pointers in the "ctypes" and "array" modules.

Break email header lines *only* on the RFC specified CR and LF characters, not on arbitrary unicode line breaks. This also fixes a bug in HTTP header parsing.

Fix email iter_attachments incorrect mutation of payload list.

Fix ssl module's parsing of GEN_RID subject alternative name fields in X.509 certs.

Remove 3DES from ssl module's default cipher list to counter measure sweet32 attack (CVE-2016-2183).

Add ChaCha20 Poly1305 to ssl module's default cipher list. (Required OpenSSL 1.1.0 or LibreSSL).

Port ssl and hashlib module to OpenSSL 1.1.0.

Remove support for passing a file descriptor to os.access. It never worked but previously didn't raise.

Fix error when distutils encounters symlink.

Fixed possible bugs when setting sqlite3.Connection.isolation_level. Based on patch by Xiang Zhang.

Fixed a crash in sqlite3.Connection.cursor() when a factory creates not a cursor. Patch by Xiang Zhang.

Avoid spurious output on OS X with Gnu Readline.

Restore deterministic behavior of random.Random().seed() for string seeds using seeding version 1. Allows sequences of calls to random() to exactly match those obtained in Python 2. Patch by Nofar Schnider.

Fix a regression in Connection.commit(). Statements should not be reset after a commit.

A new version of typing.py from <https://github.com/python/typing>: Collection (only for 3.6) (Issue #27598). Add FrozenSet to __all__ (upstream #261). Fix crash in _get_type_vars() (upstream #259). Remove the dict constraint in ForwardRef._eval_type (upstream #252).

Fix unnormalised Fraction.__pow__ result in the case of negative exponent and negative base.

cursor.description is now available for queries using CTEs.

posixpath.ismount now correctly recognizes mount points which the user does not have permission to access.

Correct some memory management errors server_hostname in _ssl.wrap_socket().

unittest.mock.create_autospec() now works properly for subclasses of property() and other data descriptors.

In the curses module, raise an error if window.getstr() or window.instr() is passed a negative value.

Fix possible usage of uninitialized memory in operator.methodcaller.

Fix possible Py_DECREF on unowned object in _sre.

Fix possible integer overflow in binascii.b2a_qp.

Fix possible integer overflow in the `_csv` module for large record lengths.

Prevent HTTPoxy attack (CVE-2016-1000110). Ignore the `HTTP_PROXY` variable when `REQUEST_METHOD` environment is set, which indicates that the script is in CGI mode.

Do not assume `sched.h` defines any `SCHED_*` constants.

In the "zlib" module, fix handling of large buffers (typically 4 GiB) when compressing and decompressing. Previously, inputs were limited to 4 GiB, and compression and decompression operations did not properly handle results of 4 GiB.

Release GIL in `nt._isdir`

Fixed unpickling by the persistent ID with protocol 0. Original patch by Alexandre Vassalotti.

Avoid an unintentional reference cycle in `email.feedparser`.

Fix error message for `imp.find_module()` to refer to 'path' instead of 'name'. Patch by Lev Maximov.

Fix SSL zero-length `recv()` calls to not block and not raise an error about unclean EOF.

Change time format returned by `http.cookie.time2netscape`, confirming the netscape cookie format and making it consistent with documentation.

Fix `activate.fish` by removing mis-use of `$`.

Fixed tracing Tkinter variables: `trace_vdelete()` with wrong mode no longer break tracing, `trace_vinfo()` now always returns a list of pairs of strings, tracing in the "u" mode now works.

Fix a scoping issue in `importlib.util.LazyLoader` which triggered an `UnboundLocalError` when lazy-loading a module that was already put into `sys.modules`.

Fixed `curses.ascii` functions `isblank()`, `isctrl()` and `ispunct()`.

Some functions (`compile()` etc) accepted a filename argument encoded as an iterable of integers. Now only strings and byte-like objects are accepted.

Prevents `distutils` failing on Windows when environment variables contain non-ASCII characters

Fixed possible leaks in the `ctypes` module.

Got rid of bare excepts in the `turtle` module. Original patch by Jelle Zijlstra.

When an exception is raised within the context being managed by a `contextlib.ExitStack()` and one of the exit stack generators catches and raises it in a chain, do not re-raise the original exception when exiting, let the new chained one through. This avoids the [PEP 479](#) bug described in [issue 25782](#).

Fix `os.urandom()` implementation using `getrandom()` on Linux. Truncate size to `INT_MAX` and loop until we collected enough random bytes, instead of casting a directly `Py_ssize_t` to `int`.

Fixed `tk.TreeView` selection operations with item id's containing spaces.

Avoid shell injection problems with `ctypes.util.find_library()`.

Fix various functions in the "readline" module to use the locale encoding, and fix `get_begidx()` and `get_endidx()` to return code point indexes.

Add `loop.connect_accepted_socket()`. Patch by Jim Fulton.

Improved behaviour of `logging.handlers.QueueListener`. Thanks to Paulo Andrade and Petr Viktorin for the analysis and patch.

Improves readability of multiprocessing error message. Thanks to Wojciech Walczak for patch.

asyncio: Set `TCP_NODELAY` by default.

Fix socket accept exhaustion during high TCP traffic. Patch by Kevin Conway.

Handle when `SO_REUSEPORT` isn't properly supported. Patch by Seth Michael Larson.

Inspect `functools.partial` in `asyncio.Handle.__repr__`. Patch by iceboy.

Fix slow pipes IO in `asyncio`. Patch by INADA Naoki.

Fix callbacks race in `asyncio.SelectorLoop.sock_connect`.

Fix selectors incorrectly retain invalid file descriptors. Patch by Mark Williams.

Refuse monitoring processes if the child watcher has no loop attached. Patch by Vincent Michel.

Raise `RuntimeError` when transport's FD is used with `add_reader`, `add_writer`, etc.

Speedup `asyncio.StreamReader.readexactly`. Patch by Štěpán Němec.

Deprecate passing `asyncio.Handle` to `run_in_executor`.

Fix `asyncio` to support formatting of non-python coroutines.

Remove UNIX socket from FS before binding. Patch by Štěpán Němec.

Prohibit Tasks to await on themselves.

Fix asyncio.Gather to refuse being cancelled once all children are done. Patch by Johannes Ebke.

Don't configure the number of workers for default threadpool executor. Initial patch by Hans Lawrenz

Optimize loop.call_soon().

Fix get_event_loop() return the current loop if called from coroutines/callbacks.

Fix inspect.isawaitable to always return bool Patch by Justin Mayfield.

Make loop methods reject socket kinds they do not support.

Fix a reflink in functools.lru_cache.

Fix asyncio.iscoroutinefunction to handle Mock objects.

Reading a corrupt config file left the parser in an invalid state. Original patch by Florian HÃ¶ch.

Fix SSL hanging if connection is closed before handshake completed. (Patch by HoHo-Ho)

Add 'interrupt execution' (^C) to Shell menu. Patch by Roger Serwy, updated by Bayard Randel.

Stop IDLE tests from 'flashing' gui widgets on the screen.

Add version to title of IDLE help window.

In section on IDLE -- console differences, mention that using exec means that __builtins__ is defined for each statement.

test_textview and test_autocomplete now pass when re-run in the same process. This occurs when test_idle fails when run with the -w option but without -jn. Fix warning from test_config.

IDLE no longer runs buggy code because of its tkinter imports. Users must include the same imports required to run directly in Python.

add line counter and crc to IDLE configHandler test dump.

Allow non-ascii chars in IDLE NEWS.txt, for contributor names.

IDLE: Cleanly delete custom themes and key bindings. Previously, when IDLE was started from a console or by import, a cascade of warnings was emitted. Patch by Serhiy Storchaka.

PyUnicode_CompareWithASCIIString() now never raises exceptions.

PyUnicode_FSDecoder() accepted a filename argument encoded as an iterable of integers. Now only strings and bytes-like objects are accepted.

Documented command-line interface of zipfile.

Disallow -j0 to be combined with -T/-l/-M in regrtest command line arguments.

Now test.support.rmtree is able to remove unwritable or unreadable directories.

Various caches now are cleared before running every test file.

regrtest: fix the parser of command line arguments.

Call gc.collect() before checking each test for "dangling threads", since the dangling threads are weak references.

In test_pyexpat, avoid testing an error message detail that changed in Expat 2.2.0.

Get Tools/scripts/fixcid.py working with Python 3 and the current "re" module, avoid invalid Python backslash escapes, and fix a bug parsing escaped C quote signs.

Fixed the type of the first argument of module-level functions generated by Argument Clinic. Patch by Petr Viktorin.

Fixed Tools/importbench/importbench.py.

Improvements to help manuals on Windows.

launcher.msi has different product codes between 32-bit and 64-bit

Ensures TargetDir is set before continuing with custom install.

Adds a shell extension to the launcher so that drag and drop works correctly.

Enabled proper Windows styles in python[w].exe manifest.

Removes hard dependency on hg.exe from PCBuild/build.bat

Added missed names to PC/python3.def.

Fix out-of-tree building on AIX. Patch by Tristan Carel and Michael Haubenwallner.

Rename --with-optimiations to --enable-optimizations.

Fix missing extensions modules when cross compiling.

Update Windows build and OS X installers to use OpenSSL 1.0.2j.

Fixed build with Estonian locale (python-config and distclean targets in Makefile). Patch by Arfrever Frehtes Taifersar Arahesis.
setup.py now detects system libffi with multiarch wrapper.

Fix the logic that searches build directories for generated include files when building outside the source tree.

Remove redundant include search directory option for building outside the source tree.

Fix clean target in freeze makefile (patch by Lisa Roach)

Update message in validate_uclibc.py

Cause lack of llvm-profdata tool when using clang as required for PGO linking to be a configure time error rather than make time when --with-optimizations is enabled. Also improve our ability to find the llvm-profdata tool on MacOS and some Linuxes.

The profile-opt build now applies PGO to the built-in modules.

Add the --with-optimizations configure flag.

Suppress spurious build warnings when updating importlib's bootstrap files. Patch by Xiang Zhang

Correct the references to Modules/python.exp and ld_so_aix, which are required on AIX. This updates references to an installation path that was changed in 3.2a4, and undoes changed references to the build tree that were made in 3.5.0a1.

CPP invocation in configure must use CPPFLAGS. Patch by Chi Hsuan Yen.

The configure script now inserts comments into the makefile to prevent the pgen and _freeze_importlib executables from being cross-compiled.

Set PYTHON_FOR_GEN in configure as the Python program to be used for file generation during the build.

Avoid C++ compilation errors on FreeBSD and OS X. Also update FreeBSD version checks for the original ctype UTF-8 workaround.

Prevent missing 'getentropy' declaration warning on macOS. Patch by Gareth Rees.