Fixed TypeError when frame.f_trace is set to None. Patch by Xavier de Gaye.

Fixed possible refleaks in failing Py_BuildValue() with the "N" format unit.

Fix possible refleak when creating a function with annotations.

Fixed bytearray.remove() for values greater than 127. Based on patch by Joe Jevnik.

int.from_bytes() no longer bypasses constructors for subclasses.

Optimized the float.fromhex() class method for exact float. It is now 2 times faster.

Single var-keyword argument of dict subtype was passed unscathed to the C-defined function. Now it is converted to exact dict.

gc.get_objects() no longer contains a broken tuple with NULL pointer.

Use RawConfigParser for .pypirc parsing, removing support for interpolation unintentionally added with move to Python 3. Behavior no longer does any interpolation in .pypirc files, matching behavior in Python 2.7 and Setuptools 19.0.

Memory functions of the :c:func:`PyMem_Malloc` domain (:c:data:`PYMEM_DOMAIN_MEM`) now use the :ref:`pymalloc allocator <pymalloc>` rather than system :c:func:`malloc`. Applications calling :c:func:`PyMem_Malloc` without holding the GIL can now crash: use `PYTHONMALLOC=debug` environment variable to validate the usage of memory allocators in your application.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main] [Misc] [NEWS.d]3.6.0a1.rst, line 94); *backlink*

Unknown interpreted text role "c:func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main] [Misc] [NEWS.d]3.6.0a1.rst, line 94); *backlink*

Unknown interpreted text role "c:data".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main] [Misc] [NEWS.d]3.6.0a1.rst, line 94); *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]3.6.0a1.rst, line 94); *backlink*

Unknown interpreted text role "c:func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main] [Misc] [NEWS.d]3.6.0a1.rst, line 94); *backlink*

Unknown interpreted text role "c:func".

Optimize function calls only using unpacking like `func(*tuple)` (no other positional argument, no keyword): avoid copying the tuple. Patch written by Joe Jevnik.

Make the builtin slice type support cycle collection.

super.__init__ no longer leaks memory if called multiple times. NOTE: A direct call of super.__init__ is not endorsed!

Fix the doc comment for FileFinder.find_spec().

Mention PEP 420 in the importlib docs.

PYTHONIOENCODING now has priority over locale in setting the error handler for stdin and stdout.

Fixed crash on iterating exhausting iterators. Affected classes are generic sequence iterators, iterators of str, bytes, bytearray, list, tuple, set, frozenset, dict, OrderedDict, corresponding views and os.scandir() iterator.

Optimize `bytes.replace(b'', b'.')` and `bytearray.replace(b'', b'.')`. Patch written by Josh Snider.

If coding cookie is specified multiple times on a line in Python source code file, only the first one is taken to account.

Add tests for reloading namespace packages.

Switch applicable importlib tests to use PEP 451 API.

Debug hooks on Python memory allocators now raise a fatal error if functions of the :c:func:`PyMem_Malloc` family are called without holding the GIL.

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

`main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, line 215); *backlink*

Unknown interpreted text role "c:func".

On error, the debug hooks on Python memory allocators now use the :mod:`tracemalloc` module to get the traceback where a memory block was allocated.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, line 225); *backlink*

Unknown interpreted text role "mod".

The debug hooks on Python memory allocator :c:func:`PyObject_Malloc` now detect when functions are called without holding the GIL.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, line 236); *backlink*

Unknown interpreted text role "c:func".

Add :envvar:`PYTHONMALLOC` environment variable to set the Python memory allocators and/or install debug hooks.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, line 246); *backlink*

Unknown interpreted text role "envvar".

The :c:func:`PyMem_SetupDebugHooks` function can now also be used on Python compiled in release mode.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, line 256); *backlink*

Unknown interpreted text role "c:func".

The :envvar:`PYTHONMALLOCSTATS` environment variable can now also be used on Python compiled in release mode. It now has no effect if set to an empty string.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, line 266); *backlink*

Unknown interpreted text role "envvar".

In debug mode, debug hooks are now also installed on Python memory allocators when Python is configured without pymalloc.

Fix str.translate() when string is ASCII and first replacements removes character, but next replacement uses a non-ASCII character or a string longer than 1 character. Regression introduced in Python 3.5.0.

Ensure exception reports from PyErr_Display() and PyErr_WriteUnraisable() are sensible even when formatting them produces secondary errors. This affects the reports produced by sys.__excepthook__() and when __del__() raises an exception.

Correct behavior to reject comma as a legal character for cookie names.

Upgrade the warning when a generator raises StopIteration from PendingDeprecationWarning to DeprecationWarning. Patch by Anish Shah.

The compiler now ignores all constant statements: bytes, str, int, float, complex, name constants (None, False, True), Ellipsis and ast.Constant; not only str and int. For example, `1.0` is now ignored in `def f(): 1.0`.

Avoid masking the original TypeError exception when using star (`*`) unpacking in function calls. Based on patch by Hagen FÃ¼rstenau and Daniel Urban.

Add a new kind of AST node: `ast.Constant`. It can be used by external AST optimizers, but the compiler does not emit directly such node.

Sped-up allocation of dict key objects by using Python's small object allocator. (Contributed by Julian Taylor.)

Import raises ImportError instead of SystemError if a relative import is attempted without a known parent package.

When compiling code, don't merge constants if they are equal but have a different types. For example, `f1, f2 = lambda: 1, lambda: 1.0` is now correctly compiled to two different functions: `f1()` returns `1` (`int`) and `f2()` returns `1.0` (`float`), even if `1` and `1.0` are equal.

The format of the `co_lnotab` attribute of code objects changes to support negative line number delta.

Add a new private _PyThreadState_UncheckedGet() function to get the current Python thread state, but don't issue a fatal error if it is NULL. This new function must be used instead of accessing directly the _PyThreadState_Current variable. The variable is no more exposed since Python 3.5.1 to hide the exact implementation of atomic C types, to avoid compiler issues.

If __package__ != __spec__.parent or if neither __package__ or __spec__ are defined then ImportWarning is raised.

[UPDATE] Comment out the one of the pickleability tests in _PyObject_GetState() due to regressions observed in Cython-based projects.

Disallowed null characters in the type name.

Fix segfault when an invalid nonlocal statement binds a name starting with two underscores.

Instances of extension types with a state that aren't subclasses of list or dict and haven't implemented any pickle-related methods (__reduce__, __reduce_ex__, __getnewargs__, __getnewargs_ex__, or __getstate__), can no longer be pickled. Including memoryview.

Massive replacing unsafe attribute setting code with special macro Py_SETREF.

Special method __bytes__() now works in str subclasses.

__sizeof__ methods of builtin types now use dynamic basic size. This allows sys.getsize() to work correctly with their subclasses with __slots__ defined.

Fixed problem with in-place string concatenation and utf-8 cache.

New Py_FinalizeEx() API allowing Python to set an exit status of 120 on failure to flush buffered streams.

telnetlib.Telnet is now a context manager.

Fixed crash in object.__reduce__() if slot name is freed inside __getattr__.

Fixed crash on converting objects with special methods __bytes__, __trunc__, and __float__ returning instances of subclasses of bytes, int, and float to subclasses of bytes, int, and float correspondingly.

Fix a possible segfault during argument parsing in functions that accept filesystem paths.

Fixed a partially broken sanity check in the _posixsubprocess internals regarding how fds_to_pass were passed to the child. The bug had no actual impact as subprocess.py already avoided it.

Fixed tokenizer crash when processing undecodable source code with a null byte.

The hash of the key now is calculated only once in most operations in C implementation of OrderedDict.

Default implementation of __reduce__ and __reduce_ex__ now rejects builtin types with not defined __new__.

Avoid buffer overreads when int(), float(), compile(), exec() and eval() are passed bytes-like objects. These objects are not necessarily terminated by a null byte, but the functions assumed they were.

Fix parser and AST: fill lineno and col_offset of "arg" node when compiling AST from Python objects.

Fixed a crash and leaking NULL in repr() of OrderedDict that was mutated by direct calls of dict methods.

Iterating OrderedDict with keys with unstable hash now raises KeyError in C implementations as well as in Python implementation.

Fixed crash when highly nested OrderedDict structures were garbage collected.

Optimize bytes.fromhex() and bytearray.fromhex(): they are now between 2x and 3.5x faster.

Optimize bytearray % args using the new private _PyBytesWriter API. Formatting is now between 2.5 and 5 times faster.

sys.setrecursionlimit() now raises a RecursionError if the new recursion limit is too low depending at the current recursion depth. Modify also the "lower-water mark" formula to make it monotonic. This mark is used to decide when the overflowed flag of the thread state is reset.

Fix input() to prompt to the redirected stdout when sys.stdout.fileno() fails.

Optimize bytes % args using the new private _PyBytesWriter API. Formatting is now up to 2 times faster.

Prevent builtin types that are not allowed to be subclassed from being subclassed through multiple inheritance.

The UTF-8 decoder is now up to 15 times as fast for error handlers: `ignore`, `replace` and `surrogateescape`.

Fixed a number of bugs in UTF-7 decoding of misformed data.

The UTF-8 encoder is now up to 75 times as fast for error handlers: `ignore`, `replace`, `surrogateescape`, `surrogatepass`. Patch co-written with Serhiy Storchaka.

Import trace messages emitted in verbose (-v) mode are no longer formatted twice.

Optimize ASCII and latin1 encoders with the `surrogateescape` error handler: the encoders are now up to 3 times as fast. Initial patch written by Serhiy Storchaka.

On Solaris 11.3 or newer, os.urandom() now uses the getrandom() function instead of the getentropy() function. The getentropy() function is blocking to generate very good quality entropy, os.urandom() doesn't need such high-quality entropy.

Modify Python's grammar to allow trailing commas in the argument list of a function declaration. For example, "def f(*, a = 3,): pass" is now legal. Patch from Mark Dickinson.

Implement PEP 498 "Literal String Interpolation". This allows you to embed expressions inside f-strings, which are converted to normal strings at run time. Given x=3, then f'value={x}' == 'value=3'. Patch by Eric V. Smith.

Fix semantic bugs when using binary operators with dictionary views and tuples.

Fix possible integer overflow and heap corruption in zipimporter.get_data().

Fix TAB key behaviour in REPL with readline.

Optimize PyLong_AsDouble.

Optimize floor and modulo division for single-digit longs. Microbenchmarks show 2-2.5x improvement. Built-in 'divmod' function is now also ~10% faster. (See also: bpo-26315)

Raise a RuntimeError when a coroutine object is awaited more than once.

Fix os.set_inheritable() on Android, ioctl() is blocked by SELinux and fails with EACCESS. The function now falls back to fcntl(). Patch written by MichaÅ, Bednarski.

Fix infinite recursion using typing.py. Thanks to Kalle Tuure!

Removed dummy methods in Tkinter widget classes: tk_menuBar() and tk_bindForTraversal().

Fix urllib.request redirect handling when the target only has a query string. Original fix by JÃ¡n Janech.

The "urllib.request" module now percent-encodes non-ASCII bytes found in redirect target URLs. Some servers send Location header fields with non-ASCII bytes, but "http.client" requires the request target to be ASCII-encodable, otherwise a UnicodeEncodeError is raised. Based on patch by Christian Heimes.

The default value of the decode_data parameter for smtpd.SMTPChannel and smtpd.SMTPServer constructors is changed to False.

Removed deprecated class asynchat.fifo.

Added readline.set_auto_history(), which can stop entries being automatically added to the history list. Based on patch by Tyler Crompton.

zipfile.ZipFile.open() can now be used to write data into a ZIP file, as well as for extracting data. Patch by Thomas Kluyver.

Honor debuglevel flag in urllib.request.HTTPHandler. Patch contributed by Chi Hsuan Yen.

In the subprocess module, allow stderr to be redirected to stdout even when stdout is not redirected. Patch by Akira Li.

mock_open 'files' no longer error on readline at end of file. Patch from Yolanda Robla.

Fixed leaking a userptr in curses panel destructor.

Removed unnecessary, and ignored, call to sum of squares helper in statistics.pvariance.

Use bisect in statistics.median instead of a linear search. Patch by Upendra Kuma.

Make use of new Decimal.as_integer_ratio() method in statistics module. Patch by Stefan Krah.

Add secrets module as described in PEP 506.

The modulefinder module now supports extended opcode arguments.

Fixed crashes related to directly created instances of types in _tkinter and curses.panel modules.

weakref.ref() no longer silently ignores keyword arguments. Patch by Georg Brandl.

xmlrpc now raises ResponseError on unsupported type tags instead of silently return incorrect result.

The __contains__ methods in the collections ABCs now check for identity before checking equality. This better matches the behavior of the concrete classes, allows sensible handling of NaNs, and makes it easier to reason about container invariants.

Fixed the comparison of plistlib.Data with other types.

Fix an uninitialized variable in *ctypes.util*. The bug only occurs on SunOS when the ctypes implementation searches for the *crle* program. Patch by Xiang Zhang. Tested on SunOS by Kees Bos.

In urllib.request, change the proxy bypass host checking against no_proxy to be case-insensitive, and to not match unrelated host names that happen to have a bypassed hostname as a suffix. Patch by Xiang Zhang.

Print server URL on http.server startup. Initial patch by Felix Kaiser.

fileinput.hook_encoded() now supports an "errors" argument for passing to open. Original patch by Joseph Hackman.

recursive_repr() now sets __qualname__ of wrapper. Patch by Xiang Zhang.

urllib.request will prefer lower_case proxy environment variables over UPPER_CASE or Mixed_Case ones. Patch contributed by Hans-Peter Jansen.

assertSequenceEqual() now correctly outputs non-stringified differing items (like bytes in the -b mode). This affects assertListEqual() and assertTupleEqual().

Remove "will be removed in Python 3.7" from deprecation messages of platform.dist() and platform.linux_distribution(). Patch by Kumaripaba Miyurusara Athukorala.

itemgetter, attrgetter and methodcaller objects no longer silently ignore keyword arguments.

Disassembling a class now disassembles class and static methods. Patch by Xiang Zhang.

Fix error handling in :func:`shutil.get_terminal_size`, catch :exc:`AttributeError` instead of :exc:`NameError`. Patch written by Emanuel Barry.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, **line 1185**); ***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]3.6.0a1.rst`, **line 1185**); ***backlink***

Unknown interpreted text role "exc".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, **line 1185**); ***backlink***

Unknown interpreted text role "exc".

tarfile's ustar and gnu formats now correctly calculate name and link field limits for multibyte character encodings like utf-8.

Fix directory traversal vulnerability with http.server on Windows. This fixes a regression that was introduced in 3.3.4rc1 and 3.4.0rc1. Based on patch by Philipp Hagemeister.

Stop encoding Latin-1-ized WSGI paths with UTF-8. Patch by Anthony Sottile.

Add STARTUPINFO to subprocess.__all__ on Windows.

Add context manager to socketserver. Patch by Aviv Palivoda.

Fix :func:`os.urandom` on Solaris 11.3 and newer when reading more than 1,024 bytes: call `getrandom()` multiple times with a limit of 1024 bytes per call.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, **line 1245**); ***backlink***

Unknown interpreted text role "func".

Eliminate http.server._quote_html() and use html.escape(quote=False). Patch by Xiang Zhang.

Raise OSError if closing a socket fails.

Add .webm to mimetypes.types_map. Patch by Giampaolo Rodola'.

Add .csv to mimetypes.types_map. Patch by Geoff Wilson.

the site module now allows .pth files to specify files to be added to sys.path (e.g. zip files).

Introduce contextlib.AbstractContextManager and typing.ContextManager.

Fixed Y2038 problem in loading binary PLists.

Handle terminal resizing with Readline 6.3+ by installing our own SIGWINCH handler. Patch by Eric Price.

Change SSLSocket.sendall() to return None, as explicitly documented for plain socket objects. Patch by Aviv Palivoda.

In http.server, respond with "413 Request header fields too large" if there are too many header fields to parse, rather than killing the connection and raising an unhandled exception. Patch by Xiang Zhang.

Added missing XMLPullParser to ElementTree.__all__.

Change BufferedReader.writable() and BufferedWriter.readable() to always return False.

Exhausted iterator of array.array now conforms with the behavior of iterators of other mutable sequences: it lefts exhausted even if iterated array is extended.

doctest.DocFileTest and doctest.testfile() now support packages (module splitted into multiple directories) for the package parameter.

Fix a regression in mock.MagicMock. _Call is a subclass of tuple (changeset 3603bae63c13 only works for classes) so we need to implement __ne__ ourselves. Patch by Andrew Plummer.

Raise ValueError rather than SystemError when a negative length is passed to SSLSocket.recv() or read().

Fix SSL recv(0) and read(0) methods to return zero bytes instead of up to 1024.

Fixed a bug in datetime.astimezone() method.

The :mod:`importlib` module now emits an :exc:`ImportError` rather than a :exc:`TypeError` if :func:`__import__` is tried during the Python shutdown process but :data:`sys.path` is already cleared (set to `None`).

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, line 1432); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, line 1432); *backlink*

Unknown interpreted text role "exc".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, line 1432); *backlink*

Unknown interpreted text role "exc".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, line 1432); *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]3.6.0a1.rst`, line 1432); *backlink*

Unknown interpreted text role "data".

:func:`warnings.formatwarning` now catches exceptions when calling :func:`linecache.getline` and :func:`tracemalloc.get_object_traceback` to be able to log :exc:`ResourceWarning` emitted late during the Python shutdown process.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, line 1443); *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]3.6.0a1.rst`, line 1443); *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]3.6.0a1.rst`, line 1443); *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]3.6.0a1.rst`, line 1443); *backlink*

Unknown interpreted text role "exc".

On Windows, faulthandler.enable() now also installs an exception handler to dump the traceback of all Python threads on any Windows exception, not only on UNIX signals (SIGSEGV, SIGFPE, SIGABRT).

Add C functions :c:func:`_PyTraceMalloc_Track` and :c:func:`_PyTraceMalloc_Untrack` to track memory blocks using the :mod:`tracemalloc` module. Add :c:func:`_PyTraceMalloc_GetTraceback` to get the traceback of an object.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, line 1466); *backlink*

Unknown interpreted text role "c:func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, **line 1466);** ***backlink***

Unknown interpreted text role "c:func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, **line 1466);** ***backlink***

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, **line 1466);** ***backlink***

Unknown interpreted text role "c:func".

The _tracemalloc now supports tracing memory allocations of multiple address spaces (domains).

Ctrl+C during Readline history search now cancels the search mode when compiled with Readline 7.

Implement a safe finalizer for the _socket.socket type. It now releases the GIL to close the socket.

spwd.getspnam() now raises a PermissionError if the user doesn't have privileges.

Avoid potential ValueError in BaseHandler.start_response. Initial patch by Peter Inglesby.

Add a new function :c:func:`PyErr_ResourceWarning` function to pass the destroyed object. Add a *source* attribute to :class:`warnings.WarningMessage`. Add warnings._showwarnmsg() which uses tracemalloc to get the traceback where source object was allocated.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, **line 1528);** ***backlink***

Unknown interpreted text role "c:func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, **line 1528);** ***backlink***

Unknown interpreted text role "class".

ssl.py _load_windows_store_certs fails if windows cert store is empty. Patch by Baji.

Fix :func:`pyclbr.readmodule` and :func:`pyclbr.readmodule_ex` to support importing packages.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\[cpython-main][Misc][NEWS.d]3.6.0a1.rst`, **line 1551);** ***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]3.6.0a1.rst`, **line 1551);** ***backlink***

Unknown interpreted text role "func".

Account for remaining Content-Length in HTTPResponse.readline() and read1(). Based on patch by Silent Ghost. Also document that HTTPResponse now supports these methods.

Handle sockets in directories unittest discovery is scanning. Patch from Victor van den Elzen.

cookiejar.http2time() now returns None if year is higher than datetime.MAXYEAR.

Fixes platform module detection of Windows Server

Fixed parsing time in week 0 before Jan 1. Original patch by TamÃ¡s Bence Gedai.

Add Mock.assert_called() and Mock.assert_called_once() methods to unittest.mock. Patch written by Amit Saha.

Invoking Path.owner() and Path.group() on Windows now raise NotImplementedError instead of ImportError.

Fixed the keys() method for Canvas and Scrollbar widgets.

Got rid of excessive buffering in fileinput. The bufsize parameter is now deprecated and ignored.

Added an optional argument timespec to the datetime isoformat() method to choose the precision of the time component.

Fix UnboundLocalError in AbstractDigestAuthHandler.get_algorithm_impls. Initial patch by Mathieu Dupuy.

Minimized overhead in copy.copy() and copy.deepcopy(). Optimized copying and deepcopying bytearrays, NotImplemented, slices, short lists, tuples, dicts, sets.

Fixed pickling and copying the accumulate() iterator with total is None.

Fixed debugging output for regular expressions with the (?x) flag.

Allowed pickling recursive dequeues.

Make mmap.write() return the number of bytes written like other write methods. Patch by Jakub Stasiak.

Fixed the subnets() methods in IP network classes for the case when resulting prefix length is equal to maximal prefix length. Based on patch by Xiang Zhang.

Remove the file if the internal open() call in NamedTemporaryFile() fails. Patch by Silent Ghost.

Fix XML-RPC client to retry when the server shuts down a persistent connection. This was a regression related to the new http.client.RemoteDisconnected exception in 3.5.0a4.

Leading `<~` is optional now in base64.a85decode() with adobe=True. Patch by Swati Jaiswal.

Remove an invalid type check in importlib.util.LazyLoader.

importlib.__import__() raises ImportError like builtins.__import__() when `level` is specified but without an accompanying package specified.

In the "socketserver" module, shut down the request (closing the connected socket) when verify_request() returns false. Patch by Aviv Palivoda.

Change the socketserver module to only catch exceptions raised from a request handler that are derived from Exception (instead of BaseException). Therefore SystemExit and KeyboardInterrupt no longer trigger the handle_error() method, and will now to stop a single-threaded server.

On Windows open the cert store readonly in ssl.enum_certificates.

os.walk() no longer uses FDs proportional to the tree depth.

Added the close() method and the support of the context manager protocol for the os.scandir() iterator.

multiprocessing: make MapResult not fail-fast upon exception.

Support keyword arguments to zlib.compress(). Patch by Aviv Palivoda.

The os.scandir() iterator now closes file descriptor not only when the iteration is finished, but when it was failed with error.

__dict__ for an OrderedDict instance is now created only when needed.

Restored support of bytes paths in os.walk() on Windows.

Add UTF-8 suggestion to error message when posting a non-Latin-1 string with http.client.

Added zipfile.ZipInfo.from_file() and zipinfo.ZipInfo.is_dir(). Patch by Thomas Kluyver.

Reset FancyURLopener's redirect counter even if there is an exception. Based on patches by Brian Brazil and Daniel Rocco.

Fixed a crash when unpickle the functools.partial object with wrong state. Fixed a leak in failed functools.partial constructor. "args" and "keywords" attributes of functools.partial have now always types tuple and dict correspondingly.

copy.deepcopy() now correctly copies range() objects with non-atomic attributes.

Path.glob() now raises a ValueError if it's called with an invalid pattern. Patch by Thomas Nyberg.

Fixed possible integer overflows in zipimport.

On Windows, getnameinfo(), gethostbyaddr() and gethostbyname_ex() functions of the socket module now decode the hostname from the ANSI code page rather than UTF-8.

The site module now writes an error into stderr if sitecustomize module can be imported but executing the module raise an ImportError. Same change for usercustomize.

xmlrpc now works with strings not encodable with used non-UTF-8 encoding.

Garbage collector now breaks reference loops with OrderedDict.

Fixed AttributeError in msilib.Directory.glob().

Added compatibility with broken protocol 2 pickles created in old Python 3 versions (3.4.3 and lower).

Deprecated accepting non-integers in grp.getgrgid().

Use cross-compilation by default for 64-bit Windows.

Add docstrings to the fields of urllib.parse results. Patch contributed by Swati Jaiswal.

Convert trace module option parsing mechanism to argparse. Patch contributed by SilentGhost.

Fix sysconfig._parse_makefile not expanding ${} vars appearing before $() vars.

Remove the deprecated apis in the trace module.

Fix mock.patch behavior when patching descriptors. Restore original values after patching. Patch contributed by Sean McCully.

In the ssl module, enable the SSL_MODE_RELEASE_BUFFERS mode option if it is safe to do so.

Don't traverse into symlinks for ** pattern in pathlib.Path.[r]glob().

Ignore PermissionError when traversing a tree with pathlib.Path.[r]glob(). Patch by Ulrich Petri.

Accept] characters in the data portion of imap responses, in order to handle the flags with square brackets accepted and produced by servers such as gmail.

fileinput now uses sys.stdin as-is if it does not have a buffer attribute (restores backward compatibility).

Optimized creating Fractions from floats by 2 times and from Decimals by 3 times.

Document as deprecated the remaining implementations of importlib.abc.Loader.load_module().

Add Decimal.as_integer_ratio().

Copying the lru_cache() wrapper object now always works, independently from the type of the wrapped object (by returning the original object unchanged).

Have the functions in compileall return booleans instead of ints and add proper documentation and tests for the return values.

Fixed possible use after free in ElementTree.XMLPullParser.

os.fwalk() no longer skips remaining directories when error occurs. Original patch by Samson Lee.

Fixed and simplified OrderedDict.__sizeof__.

Optimized deepcopying ElementTree; it is now 20 times faster.

Optimized iterating ElementTree. Iterating elements Element.iter() is now 40% faster, iterating text Element.itertext() is now up to 2.5 times faster.

Fixed various refcount issues in ElementTree iteration.

The TarFile iterator is reimplemented using generator. This implementation is simpler that using class.

Optimized ElementTree.iterparse(); it is now 2x faster. Optimized ElementTree parsing; it is now 10% faster.

Improved detecting errors in broken pickle data.

Restore the previous behaviour of tolerating most fstat() errors when opening files. This was a regression in 3.5a1, and stopped anonymous temporary files from working in special cases.

Fix regression in number of arguments compileall accepts when '-d' is specified. The check on the number of arguments has been dropped completely as it never worked correctly anyway.

In the subprocess module, preserve any exception caused by fork() failure when preexec_fn is used.

Tweak the exception message for importlib.util.resolve_name() when 'package' isn't specified but necessary.

_strptime's regexp cache now is reset after changing timezone with time.tzset().

When executing a package with the "python -m package" option, and package initialization fails, a proper traceback is now reported. The "runpy" module now lets exceptions from package initialization pass back to the caller, rather than raising ImportError.

Also in runpy and the "-m" option, omit the irrelevant message ". . . is a package and cannot be directly executed" if the package could not even be initialized (e.g. due to a bad `*.pyc` file).

Fixed problem with the mean of very small and very large numbers. As a side effect, statistics.mean and statistics.variance should be significantly faster.

Fixed copying object with state with boolean value is false.

Fixed deep copying of minidom documents. Based on patch by Marian Ganisin.

dir() on ElementTree.Element now lists properties: "tag", "text", "tail" and "attrib". Original patch by Santoso Wijaya.

Fixed a reference leak in pickle.loads() when unpickling invalid data including tuple instructions.

In the Readline completer, avoid listing duplicate global names, and search the global namespace before searching builtins.

Fixed file leak in ElementTree.iterparse() raising an error.

Fixed SystemError raised by unpickler on broken pickle data.

Fixed crash on deleting ElementTree.Element attributes.

ZipFile now always writes a ZIP_STORED header for directory entries. Patch by Dingyuan Wang.

Change three zlib functions to accept sizes that fit in Py_ssize_t, but internally cap those sizes to UINT_MAX. This resolves a regression in 3.5 where GzipFile.read() failed to read chunks larger than 2 or 4 GiB. The change affects the zlib.Decompress.decompress() max_length parameter, the zlib.decompress() bufsize parameter, and the zlib.Decompress.flush() length parameter.

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

Change semantics of EventLoop.stop() in asyncio.

When we know a subprocess.Popen process has died, do not allow the send_signal(), terminate(), or kill() methods to do anything as they could potentially signal a different process.

Added missing APIs to __all__ to match the documented APIs for the following modules: calendar, csv, enum, fileinput, ftplib, logging, optparse, tarfile, threading and wave. Also added a test.support.check__all__() helper. Patches by Jacek KoÅ‚odziej, Mauro S. M. Rodrigues and Joel Taddei.

In the Readline completer, only call getattr() once per attribute. Also complete names of attributes such as properties and slots which are listed by dir() but not yet created on an instance.

Fix a crash when garbage-collecting ctypes objects created by wrapping a memoryview. This was a regression made in 3.5a1. Based on patch by Eryksun.

Added "escape" to the __all__ list in the glob module.

Fixed recursive glob() with patterns starting with `**`.

Fix regression in smtplib's AUTH LOGIN support.

Fix the pydoc web server's module search function to handle exceptions from importing packages.

Got rid of circular references in regular expression parsing.

Command-line interface of the calendar module now uses argparse instead of optparse.

fileinput.FileInput.readline() now returns b'' instead of '' at the end if the FileInput was opened with binary mode. Patch by Ryosuke Ito.

Fixed inspect.getdoc() for inherited docstrings of properties. Original patch by John Mark Vandenberg.

Always use os.urandom as a source of randomness in uuid.uuid4.

Fixed textwrap.dedent() for the case when largest common whitespace is a substring of smallest leading whitespace. Based on patch by Robert Li.

The lru_cache() wrapper objects now can be copied and pickled (by returning the original object unchanged).

typing: Don't crash on Union[str, Pattern].

asyncio: Raise error from drain() when socket is closed.

Cleaned up and fixed minor bugs in C implementation of OrderedDict.

Improved Unicode support in SMTPHandler through better use of the email package. Thanks to user simon04 for the patch.

Move the imp module from a PendingDeprecationWarning to DeprecationWarning.

Remove mentions of the formatter module being removed in Python 3.6.

Fixed a bug in C implementation of OrderedDict.move_to_end() that caused segmentation fault or hang in iterating after moving several items to the start of ordered dict.

pickletools.dis() now outputs implicit memo index for the MEMOIZE opcode.

Add an optional newline parameter to binascii.b2a_base64(). base64.b64encode() uses it to avoid a memory copy.

Objects that need calling `__new__` with keyword arguments, can now be pickled using pickle protocols older than protocol version 4.

zipfile now works in threads disabled builds.

smtpd's SMTPChannel now correctly raises a ValueError if both decode_data and enable_SMTPUTF8 are set to true.

RobotFileParser now supports Crawl-delay and Request-rate extensions. Patch by Nikolay Bogoychev.

distutils raises OSError instead of DistutilsPlatformError when MSVC is not installed.

Fixed protocol for the STACK_GLOBAL opcode in pickletools.opcodes.

Updates asyncio datagram create method allowing reuseport and reuseaddr socket options to be set prior to binding the socket. Mirroring the existing asyncio create_server method the reuseaddr option for datagram sockets defaults to True if the O/S is 'posix' (except if the platform is Cygwin). Patch by Chris Laws.

Add asyncio.run_coroutine_threadsafe(). This lets you submit a coroutine to a loop from another thread, returning a concurrent.futures.Future. By Vincent Michel.

Fix CGIRequestHandler to split the query from the URL at the first question mark (?) rather than the last. Patch from Xiang Zhang.

Prevent CGIRequestHandler from collapsing slashes in the query part of the URL as if it were a path. Patch from Xiang Zhang.

Don't add crypt.METHOD_CRYPT to crypt.methods if it's not supported. Check if it is supported, it may not be supported on OpenBSD for example.

Default implementation of tzinfo.fromutc() was returning wrong results in some cases.

Failed readline.set_completer_delims() no longer left the module in inconsistent state.

rlcompleter now omits private and special attribute names unless the prefix starts with underscores.

rlcompleter now can add a space or a colon after completed keyword.

timezone.utc name is now plain 'UTC', not 'UTC-00:00'.

fromtimestamp() and utcfromtimestamp() methods of datetime.datetime now round microseconds to nearest with ties going to nearest even integer (ROUND_HALF_EVEN), as round(float), instead of rounding towards -Infinity (ROUND_FLOOR).

Timeit now warns when there is substantial (4x) variance between best and worst times. Patch from Serhiy Storchaka.

site-packages/README -> README.txt.

help() and pydoc can now list named tuple fields in the order they were defined rather than alphabetically. The ordering is determined by the _fields attribute if present.

Improve speed of itertools.cycle() and make its pickle more compact.

Fix crash in itertools.cycle.__setstate__() when the first argument wasn't a list.

urllib.parse raises ValueError on all invalid ports. Patch by Martin Panter.

Improve __repr__ of argparse.Namespace() for invalid identifiers. Patch by Matthias Bussonnier.

run_setup was broken in distutils. Patch from Alexander Belopolsky.

2to3 converts StringTypes to a tuple. Patch from Mark Hammond.

open() accepted a 'U' mode string containing '+', but 'U' can only be used with 'r'. Patch from Jeff Balogh and John O'Connor.

improved tests for zipimporter2. Patch from Mark Lawrence.

unittest.mock.mock_open().reset_mock would recurse infinitely. Patch from Nicola Palumbo and Laurent De Buyst.

Fast searching optimization in regular expressions now works for patterns that starts with capturing groups. Fast searching optimization now can't be disabled at compile time.

unittest.mock side_effects can now be exceptions again. This was a regression vs Python 3.4. Patch from Ignacio Rossi

Remove deprecated inspect.getmoduleinfo function.

Fix (another) memory leak in SSLSocket.getpeercer().

Disable the vulnerable SSLv3 protocol by default when creating ssl.SSLContext.

Fix memory leak in SSLSocket.getpeercert().

Sockets returned from accept() shouldn't appear to be nonblocking.

When threading.Event is reinitialized, the underlying condition should use a regular lock rather than a recursive lock.

Skip getaddrinfo if host is already resolved. Patch by A. Jesse Jiryu Davis.

Add asyncio.StreamReader.readuntil() method. Patch by ĞœĞ°Ñ€Ğº ĞšĞ¾Ñ€ĞµĞ½Ğ±ĞµÑ€Ğ³.

Avoid unnecessary serialization of getaddrinfo(3) calls on OS X versions 10.5 or higher. Original patch by A. Jesse Jiryu Davis.

Avoid unnecessary serialization of getaddrinfo(3) calls on current versions of OpenBSD and NetBSD. Patch by A. Jesse Jiryu Davis.

Fix asyncio/subprocess.communicate() to handle empty input. Patch by Jack O'Connor.

Add loop.get_exception_handler method

asyncio: Add loop.create_future method

Add tests for idlelib.configHelpSourceEdit. Patch by Saimadhav Heblikar.

In the 'IDLE-console differences' section of the IDLE doc, clarify how running with IDLE affects sys.modules and the standard streams.

fix incorrect change in IOBinding that prevented printing. Augment IOBinding htest to include all major IOBinding functions.

Revert unwanted conversion of ' to â€™ RIGHT SINGLE QUOTATION MARK in README.txt and open this and NEWS.txt with 'ascii'. Re-encode CREDITS.txt to utf-8 and open it with 'utf-8'.

Stop the debugger engine (normally in a user process) before closing the debugger window (running in the IDLE process). This prevents the RuntimeErrors that were being caught and ignored.

Prevent IDLE from hanging when a) closing the shell while the debugger is active (15347); b) closing the debugger with the [X] button (15348); and c) activating the debugger when already active (24455). The patch by Mark Roseman does this by making two changes. 1. Suspend and resume the gui.interaction method with the tcl vwait mechanism intended for this purpose (instead of root.mainloop & .quit). 2. In gui.run, allow any existing interaction to terminate first.

Change 'The program' to 'Your program' in an IDLE 'kill program?' message to make it clearer that the program referred to is the currently running user program, not IDLE itself.

Improve the appearance of the IDLE editor window status bar. Patch by Mark Roseman.

Change the handling of new built-in text color themes to better address the compatibility problem introduced by the addition of IDLE Dark. Consistently use the revised idleConf.CurrentTheme everywhere in idlelib.

Extension configuration is now a tab in the IDLE Preferences dialog rather than a separate dialog. The former tabs are now a sorted list. Patch by Mark Roseman.

Re-activate the config dialog help button with some content about the other buttons and the new IDLE Dark theme.

IDLE now has an 'IDLE Dark' built-in text color theme. It is more or less IDLE Classic inverted, with a cobalt blue background. Strings, comments, keywords, ... are still green, red, orange, ... To use it with IDLEs released before November 2015, hit the 'Save as New Custom Theme' button and enter a new name, such as 'Custom Dark'. The custom theme will work with any IDLE release, and can be modified.

README.txt is now an idlelib index for IDLE developers and curious users. The previous user content is now in the IDLE doc chapter. 'IDLE' now means 'Integrated Development and Learning Environment'.

Users can now set breakpoint colors in Settings -> Custom Highlighting. Original patch by Mark Roseman.

Inactive selection background now matches active selection background, as configured by users, on all systems. Found items are now always highlighted on Windows. Initial patch by Mark Roseman.

Idle: make calltip and completion boxes appear on Macs affected by a tk regression. Initial patch by Mark Roseman.

Idle ScrolledList context menus (used in debugger) now work on Mac Aqua. Patch by Mark Roseman.

Make right-click for context menu work on Mac Aqua. Patch by Mark Roseman.

Associate tkinter messageboxes with a specific widget. For Mac OSX, make them a 'sheet'. Patch by Mark Roseman.

Enhance the initial html viewer now used for Idle Help. Properly indent fixed-pitch text (patch by Mark Roseman). Give code snippet a very Sphinx-like light blueish-gray background. Re-use initial width and height set by users for shell and editor. When the Table of Contents (TOC) menu is used, put the section header at the top of the screen.

Condense and rewrite Idle doc section on text colors.

Explain some differences between IDLE and console Python.

Explain need for *print* when running file from Idle editor.

Doc: augment Idle feature list and no-subprocess section.

Update doc for Idle command line options. Some were missing and notes were not correct.

Most of idlelib is private and subject to change. Use idleib.idle.* to start Idle. See idlelib.__init__.__doc__.

Idle: add synchronization comments for future maintainers.

Replace help.txt with help.html for Idle doc display. The new idlelib/help.html is rstripped Doc/build/html/library/idle.html. It looks better than help.txt and will better document Idle as released. The tkinter html viewer that works for this file was written by Rose Roseman. The now unused EditorWindow.HelpDialog class and helt.txt file are deprecated.

Deprecate unused idlelib.idlever with possible removal in 3.6.

Remove extraneous code (which also create 2 & 3 conflicts).

Used HTTPS for external links in the documentation if possible.

Rework the Readline module documentation to group related functions together, and add more details such as what underlying Readline functions and variables are accessed.

Adds note to ctypes documentation regarding cdll.msvcrt.

Clarify the default size argument of stack_size() in the "threading" and "_thread" modules. Patch from Mattip.

Update 3.x packaging documentation: * "See also" links to the new docs are now provided in the legacy pages * links to setuptools documentation have been updated

Added tests for the turtle module. Patch by ingrid, Gregory Loyse and Jelle Zijlstra.

When using "python3 -m test --testdir=TESTDIR", regrtest doesn't add "test." prefix to test module names.

The multiprocessing thread pool (multiprocessing.dummy.Pool) was untested.

Added new tests for pickling iterators of mutable sequences.

Added test.support.check_no_resource_warning() to check that no ResourceWarning is emitted.

Changed test_ssl to use its internal local server more. This avoids relying on svn.python.org, which recently changed root certificate.

Tests for OrderedDict are extracted from test_collections into separate file test_ordered_dict.

Added tests for OrderedDict subclasses.

Add -P/--pgo to test.regrtest to suppress error output when running the test suite for the purposes of a PGO build. Initial patch by Alecsandru Patrascu.

Add `python -m test --list-tests` command to list tests.

`python -m test --huntrleaks ...` now also checks for leak of file descriptors. Patch written by Richard Oudkerk.

Fix `python -m test --coverage` on Windows. Remove the list of ignored directories.

`PCbuild\rt.bat` now accepts an unlimited number of arguments to pass along to regrtest.py. Previously there was a limit of 9.

Skip test_timestamp_overflow in test_import if bytecode files cannot be written.

Don't try to link _ctypes with a ffi_convenience library.

Fix linking extension modules for cross builds. Patch by Xavier de Gaye.

Fixed support of RTLD_* constants defined as enum values, not via macros (in particular on Android). Patch by Chi Hsuan Yen.

Disable the rules for running _freeze_importlib and pgen when cross-compiling. The output of these programs is normally saved with the source code anyway, and is still regenerated when doing a native build. Patch by Xavier de Gaye.

Link audioop, _datetime, _ctypes_test modules to libm, except on Mac OS X. Patch written by Chi Hsuan Yen.

A --with-lto configure option has been added that will enable link time optimizations at build time during a make profile-opt. Some compilers and toolchains are known to not produce stable code when using LTO, be sure to test things thoroughly before relying on it. It can provide a few % speed up over profile-opt alone.

Adds validation of ucrtbase[d].dll version with warning for old versions.

Avoid error about nonexistent fileblocks.o file by using a lower-level check for st_blocks in struct stat.

Fixing the build output folder for tix-8.4.3.6. Patch by Bjoern Thiel.

Update Windows builds to use OpenSSL 1.0.2g.

Added `--pgo` and `--pgo-job` arguments to `PCbuild\build.bat` for building with Profile-Guided Optimization. The old `PCbuild\build_pgo.bat` script is removed.

Add support for building with ICC to `configure`, including a new `--with-icc` flag.

Fix installation of Python on UNIX with make -j9.

It is now possible to build Python on Windows without errors when external libraries are not available.

Compile Modules/_math.c once, before building extensions. Previously it could fail to compile properly if the math and cmath builds were concurrent.

Update OS X 10.5+ 32-bit-only installer to build and link with OpenSSL 1.0.2g.

Update Windows builds to use OpenSSL 1.0.2f.

Support Apple Xcode 7's new textual SDK stub libraries.

Do not enable unreachable code warnings when using gcc as the option does not work correctly in older versions of gcc and has been silently removed as of gcc-4.5.

Updates make_zip.py to correctly generate library ZIP file.

Update the prepare_ssl.py script to handle OpenSSL releases that don't include the contents of the include directory (that is, 1.0.2e and later).

bdist_wininst created binaries fail to start and find 32bit Python

Update the list of magic numbers in launcher

Excludes venv from library when generating embeddable distro.

Removed very outdated PC/example_nt/ directory.

Fix python-gdb.py: don't get C types once when the Python code is loaded, but get C types on demand. The C types can change if python-gdb.py is loaded before the Python executable. Patch written by Thomas Ilsche.

Fix the Freeze tool to properly use flags passed through configure. Patch by Daniel Shaulov.

Add dictionary unpacking support to Tools/parser/unparse.py. Patch by Guo Ci Teo.

Fix variable name typo in Argument Clinic.

Fix output of python-config --extension-suffix.

The pyvenv script has been deprecated in favour of *python3 -m venv*.

SystemError is now raised in all programming bugs with using PyArg_ParseTupleAndKeywords(). RuntimeError did raised before in some programming bugs.

ValueError is now raised instead of TypeError on buffer overflow in parsing "es#" and "et#" format units. SystemError is now raised instead of TypeError on programmatical error in parsing format string.