PEP 475 - EINTR handling.

Fix many edge cases (including crashes) involving custom mro() implementations.

Avoid using PyObject_AsCharBuffer(), PyObject_AsReadBuffer() and PyObject_AsWriteBuffer().

Revert some changes (issue #16795) to AST line numbers and column offsets that constituted a regression.

Allow changing an object's __class__ between a dynamic type and static type in some cases.

PyUnicode_EncodeFSDefault(), PyUnicode_EncodeMBCS() and PyUnicode_EncodeCodePage() now raise an exception if the object is not a Unicode object. For PyUnicode_EncodeFSDefault(), it was already the case on platforms other than Windows. Patch written by Campbell Barton.

The default __ne__() now returns NotImplemented if __eq__() returned NotImplemented. Original patch by Martin Panter.

Fixed a crash in str.decode() when error handler returned replacement string longer than malformed input data.

The "backslashreplace" error handlers now works with decoding and translating.

Delay-load ShellExecute[AW] in os.startfile for reduced startup overhead on Windows.

pyatomic.h now uses stdatomic.h or GCC built-in functions for atomic memory access if available. Patch written by Vitor de Lima and Gustavo Temple.

%-interpolation (aka printf) formatting added for bytes and bytearray.

Fix jumping out of an infinite while loop in the pdb.

bytes constructor now raises TypeError when encoding or errors is specified with non-string argument. Based on patch by Renaud Blanch.

If the current working directory ends up being set to a non-existent directory then import will no longer raise FileNotFoundError.

Move the interpreter startup & shutdown code to a new dedicated pylifecycle.c module

Improve method cache efficiency.

Fix crash when trying to enlarge a bytearray to 0x7fffffff bytes on a 32-bit platform.

Fix an assertion failure in debug mode when doing a reentrant dict insertion in debug mode.

Fix integer overflow in Unicode case operations (upper, lower, title, swapcase, casefold).

Circular imports involving relative imports are now supported.

Fix assertion error in debug mode when dividing a complex number by (nan+0j).

Do not raise ImportWarning when sys.path_hooks or sys.meta_path are set to None.

Use 'bytes-like object required' in error messages that previously used the far more cryptic "'x' does not support the buffer protocol.

Fixed integer overflow issues in "backslashreplace", "xmlcharrefreplace", and "surrogatepass" error handlers.

speed up *PyObject_IsInstance* and *PyObject_IsSubclass* in the common case that the second argument has metaclass *type*.

Add a new *PyErr_FormatV* function, similar to *PyErr_Format* but accepting a *va_list* argument.

Fix overflow checking when generating the repr of a unicode object.

Fix overflow checking in PyBytes_Repr.

Fix integer overflow issues in latin-1 encoding.

_charset parameter of MIMEText now also accepts email.charset.Charset instances. Initial patch by Claude Paroz.

Fix inspect.getsource() to support decorated functions. Patch by Claudiu Popa.

os.__all__ includes posix functions.

Use os.path.abspath in the shutil module.

avoid generating a JUMP_FORWARD instruction at the end of an if-block if there is no else-clause. Original patch by Eugene Toder.

Now ValueError is raised instead of TypeError when str or bytes argument contains not permitted null character or byte.

Fix the internal function set_inheritable() on Illumos. This platform exposes the function `ioctl(FIOCLEX)`, but calling it fails with errno is ENOTTY: "Inappropriate ioctl for device". set_inheritable() now falls back to the slower `fcntl()` (`F_GETFD` and then `F_SETFD`).

Displaying the __qualname__ of the underlying function in the repr of a bound method.

Using pthread, PyThread_create_key() now sets errno to ENOMEM and returns -1 (error) on integer overflow.

Argument Clinic based signature introspection added for 30 of the builtin functions.

C functions and methods (of the 'builtin_function_or_method' type) can now be weakref'ed. Patch by Wei Wu.

Improve index error messages for bytearrays, bytes, lists, and tuples by adding 'or slices'. Added ', not <typename>' for bytearrays.

Original patch by Claudiu Popa.

Apply Argument Clinic to bytes and bytearray. Patch by Tal Einat.

Clear interned strings in slotdefs.

Upgrade Unicode database to Unicode 7.0.0.

Fix a crash with the f_locals attribute with closure variables when frame.clear() has been called.

Add a new `__qualname__` attribute to generator, the qualified name, and use it in the representation of a generator (`repr(gen)`). The default name of the generator (`__name__` attribute) is now get from the function instead of the code. Use `gen.gi_code.co_name` to get the name of the code.

With the aid of heuristics in SyntaxError.__init__, the parser now attempts to generate more meaningful (or at least more search engine friendly) error messages when "exec" and "print" are used as statements.

In the conditional if-else expression, allow an integer written with no space between itself and the `else` keyword (e.g. `True if 42else False`) to be valid syntax.

Fix over-pessimistic computation of the stack effect of some opcodes in the compiler. This also fixes a quadratic compilation time issue noticeable when compiling code with a large number of "and" and "or" operators.

Fix a crash in the builtin function super() when called without argument and without current frame (ex: embedded Python).

Fix flushing of standard streams in the interactive interpreter.

In rare cases, when running finalizers on objects in cyclic trash a bad pointer dereference could occur due to a subtle flaw in internal iteration logic.

PyBytes_Concat() now tries to concatenate in-place when the first argument has a reference count of 1. Patch by Nikolaus Rath.

-W command line options now have higher priority than the PYTHONWARNINGS environment variable. Patch by Arfrever.

Define PATH_MAX for GNU/Hurd in Python/pythonrun.c.

Support setting FPU precision on m68k.

Fix sending tuples to custom generator objects with the yield from syntax.

pow(a, b, c) now raises ValueError rather than TypeError when b is negative. Patch by Josh Rosenberg.

PEP 465: Add the '@' operator for matrix multiplication.

Fix segfault when str is called on an uninitialized UnicodeEncodeError, UnicodeDecodeError, or UnicodeTranslateError object.

Fix PyUnicode_DATA() alignment under m68k. Patch by Andreas Schwab.

Add a type cast to avoid shifting a negative number.

Properly position in source code files even if they are opened in text mode. Patch by Serhiy Storchaka.

Key-sharing now also works for instance dictionaries of subclasses. Patch by Peter Ingebretson.

Attributes missing from modules now include the module name in the error text. Original patch by ysj.ray.

%c, %o, %x, and %X now raise TypeError on non-integer input.

The ASDL parser - used by the build process to generate code for managing the Python AST in C - was rewritten. The new parser is self contained and does not require to carry long the spark.py parser-generator library; spark.py was removed from the source base.

Allow `\x00` to be used as a fill character when using str, int, float, and complex __format__ methods.

Add ipaddress.reverse_pointer. Patch by Leon Weber.

Modify string.Formatter to support auto-numbering of replacement fields. It now matches the behavior of str.format() in this regard. Patches by Phil Elson and Ramchandra Apte.

Make alternate formatting ('#') for type 'c' raise an exception. In versions prior to 3.5, '#' with 'c' had no effect. Now specifying it is an error. Patch by Torsten Landschoff.

Perform overflow checks before allocating memory in the _Py_char2wchar function.

pyvenv creates relative symlinks where possible.

cgi.FieldStorage() now supports the context management protocol.

Print response headers for CONNECT requests when debuglevel > 0. Patch by Demian Brecht.

Optimized io.BytesIO to make less allocations and copyings.

Splitting on a pattern that could match an empty string now raises a warning. Patterns that can only match empty strings are now rejected.

Closing io.BytesIO with exported buffer is rejected now to prevent corrupting exported buffer.

Removed __ne__ implementations. Since fixing default __ne__ implementation in issue #21408 they are redundant.

Fix possible overflow in itertools.permutations.

Fix possible overflow in itertools.product.

Fixed possible integer overflow in itertools.combinations.

Fixed possible integer overflow in _json.encode_basestring_ascii.

Fix the exception handling of generators in PyEval_EvalFrameEx(). At entry, save or swap the exception state even if PyEval_EvalFrameEx() is called with throwflag=0. At exit, the exception state is now always restored or swapped, not only if why is WHY_YIELD or WHY_RETURN. Patch co-written with Antoine Pitrou.

Restored support of writing ZIP files to tellable but non-seekable streams.

Writing to ZipFile and reading multiple ZipExtFiles is threadsafe now.

JSON decoder now raises JSONDecodeError instead of ValueError.

timeit now rejects statements which can't be compiled outside a function or a loop (e.g. "return" or "break").

Fixed readline with frames in Python implementation of pickle.

Fixed bugs in the comparison of ipaddress classes.

Removed incorrect implementations of __ne__() which didn't returned NotImplemented if __eq__() returned NotImplemented. The default __ne__() now works correctly.

:class:`email.feedparser.FeedParser` now handles (malformed) headers with no key rather than assuming the body has started.

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.5.0a1.rst`, line 913); *backlink***
>
> Unknown interpreted text role "class".

Support Application-Layer Protocol Negotiation (ALPN) in the ssl module.

Pickling of ipaddress objects now produces more compact and portable representation.

Update ssl error codes from latest OpenSSL git master.

Much faster implementation of ipaddress.collapse_addresses() when there are many non-consecutive addresses.

64-bit dev_t is now supported in the os module.

When an exception is raised in a task submitted to a ProcessPoolExecutor, the remote traceback is now displayed in the parent process. Patch by Claudiu Popa.

Add an option to limit output size when decompressing LZMA data. Patch by Nikolaus Rath and Martin Panter.

In the http.cookies module, capitalize "HttpOnly" and "Secure" as they are written in the standard.

In the distutils' check command, fix parsing of reST with code or code-block directives.

selectors.BaseSelector.get_key() now raises a RuntimeError if the selector is closed. And selectors.BaseSelector.close() now clears its internal reference to the selector mapping to break a reference cycle. Initial patch written by Martin Richard. (See also: bpo-23225)

Provide a way to seed the linecache for a PEP-302 module without actually loading the code.

Provide a new object API for traceback, including the ability to not lookup lines at all until the traceback is actually rendered, without any trace of the original objects being kept alive.

Provide a home() classmethod on Path objects. Contributed by Victor Salgado and Mayank Tripathi.

Make `json.dumps(..., ensure_ascii=False)` as fast as the default case of `ensure_ascii=True`. Patch by Naoki Inada.

Add math.inf and math.nan constants.

Add ssl.SSLObject.shared_ciphers() and ssl.SSLSocket.shared_ciphers() to fetch the client's list ciphers sent at handshake.

Remove compatibility with OpenSSLs older than 0.9.8.

Improve performance and introspection support of comparison methods created by functool.total_ordering.

Add an expanduser() method on Path objects.

Fix SimpleHTTPServer to correctly carry the query string and fragment when it redirects to add a trailing slash.

Added http.HTTPStatus enums (i.e. HTTPStatus.OK, HTTPStatus.NOT_FOUND). Patch by Demian Brecht.

In the io, module allow more operations to work on detached streams.

In the ftplib, make ssl.PROTOCOL_SSLv23 the default protocol version.

On OpenBSD 5.6 and newer, os.urandom() now calls getentropy(), instead of reading /dev/urandom, to get pseudo-random bytes.

pprint now produces evaluable output for wrapped strings.

Added missing names to codecs.__all__. Patch by Martin Panter.

Pickling now uses the NEWOBJ opcode instead of the NEWOBJ_EX opcode if possible.

Added a __sizeof__ implementation for pickle classes.

pickletools.optimize() now aware of the MEMOIZE opcode, can produce more compact result and no longer produces invalid output if input data contains MEMOIZE opcodes together with PUT or BINPUT opcodes.

Fixed HTTPConnection.set_tunnel with default port. The port value in the host header was set to "None". Patch by Demian Brecht.

A warning no longer produces an AttributeError when the program is run with pythonw.exe.

shutil.copytree(): fix crash when copying to VFAT. An exception handler assumed that OSError objects always have a 'winerror' attribute. That is not the case, so the exception handler itself raised AttributeError when run on Linux (and, presumably, any other non-Windows OS). Patch by Greg Ward.

Fix inspect.getsource() to load updated source of reloaded module. Initial patch by Berker Peksag.

Support wrapped callables in doctest. Patch by Claudiu Popa.

Make sure selectors.EpollSelector.select() works when no FD is registered.

In the constructor of http.client.HTTPSConnection, prefer the context's check_hostname attribute over the *check_hostname* parameter.

Add function :func:`sys.is_finalizing` to know about interpreter shutdown.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.5.0a1.rst`, line 1277); *backlink*
>
> Unknown interpreted text role "func".

Add a default limit for the amount of data xmlrpclib.gzip_decode will return. This resolves CVE-2013-1753.

ZipFile.open() no longer reopen the underlying file. Objects returned by ZipFile.open() can now operate independently of the ZipFile even if the ZipFile was created by passing in a file-like object as the first argument to the constructor.

Fix __pycache__ pyc file name clobber when pyc_compile is asked to compile a source file containing multiple dots in the source file name.

Update turtledemo doc and add module to the index.

Fixed socket leak if HTTPConnection.getresponse() fails. Original patch by Martin Panter.

Deprecated the use of re.LOCALE flag with str patterns or re.ASCII. It was newer worked.

The "ip" command is now used on Linux to determine MAC address in uuid.getnode(). Pach by Bruno Cauet.

Add a context argument to xmlrpclib.ServerProxy constructor.

Add contextlib.redirect_stderr().

Make ssl.RAND_egd() optional to support LibreSSL. The availability of the function is checked during the compilation. Patch written by Bernard Spil.

SAX parser now supports files opened with file descriptor or bytes path.

Constructors and update methods of mapping classes in the collections module now accept the self keyword argument.

Add readline.append_history_file.

Added the "namereplace" error handler.

Add *context* parameter to logging.handlers.HTTPHandler.

Allow SSLContext to take the *hostname* parameter even if OpenSSL doesn't support SNI.

TestCase.subTest() would cause the test suite to be stopped when in failfast mode, even in the absence of failures.

HTTP cookie parsing is now stricter, in order to protect against potential injection attacks.

Windows detection in pathlib is now more robust.

Reject coroutines in asyncio add_signal_handler(). Patch by Ludovic.Gasc.

Added urllib.request.HTTPBasicPriorAuthHandler. Patch by Matej Cepl.

Added attributes to the re.error class.

Fix possible double free in the io.TextIOWrapper constructor.

Different Unicode characters having the same uppercase but different lowercase are now matched in case-insensitive regular expressions.

Fixed fcntl() with integer argument on 64-bit big-endian platforms.

Add an *--sort-keys* option to json.tool CLI.

Updated reprlib output format for sets to use set literals. Patch contributed by Berker Peksag.

Updated reprlib output format for arrays to display empty arrays without an unnecessary empty list. Suggested by Serhiy Storchaka.

Fixed the uu_codec codec incorrectly ported to 3.x. Based on patch by Martin Panter.

uuid.getnode() now determines MAC address on AIX using netstat. Based on patch by Aivars Kalvāns.

Fixed ttk.Treeview.tag_has() when called without arguments.

Verify certificates by default in httplib (PEP 476).

Fixed unpickling of http.cookies.SimpleCookie with protocol 2 and above. Patch by Tim Graham.

Brought excluded code into the scope of a try block in SysLogHandler.emit().

Add missing get_terminal_size and SameFileError to shutil.__all__.

Remove deprecated Netrc class in the ftplib module. Patch by Matt Chaput.

Fixed handling of case-insensitive ranges in regular expressions.

Module level functions in the re module now cache compiled locale-dependent regular expressions taking into account the locale.

Query methods on pathlib.Path() (exists(), is_dir(), etc.) now return False when the underlying stat call raises NotADirectoryError.

distutils now falls back to copying files when hard linking doesn't work. This allows use with special filesystems such as VirtualBox shared folders.

Implemented reprs of classes in the zipfile module.

Honour load_tests in the start_dir of discovery.

gettext now raises an error when a .mo file has an unsupported major version number. Patch by Aaron Hill.

Provide a locale.delocalize() function which can remove locale-specific number formatting from a string representing a number, without then converting it to a specific type. Patch by Cédric Krier.

Make the pickling of global objects which don't have a __module__ attribute less slow.

Fixed ResourceWarning in shlex.__nain__.

Defaults set with set_defaults on an argparse subparser are no longer ignored when also set on the parent parser.

unittest test loading ImportErrors are reported as import errors with their import exception rather than as attribute errors after the import has already failed.

Make it possible to examine the errors from unittest discovery without executing the test suite. The new *errors* attribute on TestLoader exposes these non-fatal errors encountered during discovery.

Make email.headerregistry's header 'params' attributes be read-only (MappingProxyType). Previously the dictionary was modifiable but a new one was created on each access of the attribute.

SSLv3 is now disabled throughout the standard library. It can still be enabled by instantiating a SSLContext manually.

In asyncio, the default SSL context for client connections is now created using ssl.create_default_context(), for stronger security.

Include closefd in io.FileIO repr.

Add silent mode for compileall. quiet parameters of compile_{dir, file, path} functions now have a multilevel value. Also, -q option of the CLI now have a multilevel value. Patch by Thomas Kluyver.

Convert the array and cmath modules to Argument Clinic.

Add socket.socketpair() on Windows.

Fix a file descriptor leak when socketserver bind fails.

Fixed segfault in CTypes POINTER handling of large values.

Raise ConversionError in xdrlib as documented. Patch by Filip Gruszczyński and Claudiu Popa.

Optimized parsing of regular expressions.

Now unmatched groups are replaced with empty strings in re.sub() and re.subn().

sndhdr.what/whathdr now return a namedtuple.

Fix pyexpat's creation of a dummy frame to make it appear in exception tracebacks.

Add support for in-memory SSL to the ssl module. Patch by Geert Jansen.

Fix len() on a WeakKeyDictionary when .clear() was called with an iterator alive.

Eliminated race condition in the computation of names for new threads.

Avoid RuntimeError in pickle.whichmodule() when sys.modules is mutated while iterating. Patch by Olivier Grisel.

concurrent.futures.Executor.map() now takes a *chunksize* argument to allow batching of tasks in child processes and improve performance of ProcessPoolExecutor. Patch by Dan O'Reilly.

os.path.join() and os.path.relpath() now raise a TypeError with more helpful error message for unsupported or mismatched types of arguments.

The zipfile module CLI now adds entries for directories (including empty directories) in ZIP file.

In the ssl.SSLContext.load_default_certs, consult the environmental variables SSL_CERT_DIR and SSL_CERT_FILE on Windows.

The email.__version__ variable has been removed; the email code is no longer shipped separately from the stdlib, and __version__ hasn't been updated in several releases.

Added non derived UTF-8 aliases to locale aliases table.

Added locales supported in glibc 2.18 to locale alias table.

Added convenience methods read_text/write_text and read_bytes/ write_bytes to pathlib.Path objects.

On 32-bit AIX platform, don't expose os.posix_fadvise() nor os.posix_fallocate() because their prototypes in system headers are wrong.

When an io.BufferedRWPair object is deallocated, clear its weakrefs.

Number of capturing groups in regular expression is no longer limited by 100.

InteractiveInterpreter now displays the full chained traceback in its showtraceback method, to match the built in interactive interpreter.

Added tests for marshal C API that works with FILE*.

distutils register and upload methods now use HTML standards compliant CRLF line endings.

Fixed macpath.join() for empty first component. Patch by Oleg Oshmyan.

distutils' build and build_ext commands now accept a `-j` option to enable parallel building of extension modules.

Improve canceled timer handles cleanup to prevent unbound memory usage. Patch by Joshua Moore-Oliva.

TemporaryDirectory no longer attempts to clean up twice when used in the with statement in generator.

Forbidden ambiguous octal escapes out of range 0-0o377 in regular expressions.

Now directories added to ZIP file have correct Unix and MS-DOS directory attributes.

ZipFile.close() no longer writes ZIP64 central directory records if allowZip64 is false.

Fix urljoin problem with relative urls, a regression observed after changes to issue22118 were submitted.

Fixed debugging output of the GROUPREF_EXISTS opcode in the re module. Removed trailing spaces in debugging output.

Unhandled exception in thread no longer causes unhandled AttributeError when sys.stderr is None.

Ensure that `bufsize=1` in subprocess.Popen() selects line buffering, rather than block buffering. Patch by Akira Li.

Fix API bug: email.message.EmailMessage.is_attachment is now a method.

Fix email.message.EmailMessage.is_attachment to return the correct result when the header has parameters as well as a value.

Add NNTPError to nntplib.__all__.

urllib.request.urlopen will accept a context object (SSLContext) as an argument which will then be used for HTTPS connection. Patch by Alex Gaynor.

The warnings registries are now reset when the filters are modified.

Limit the length of incoming HTTP request in wsgiref server to 65536 bytes and send a 414 error code for higher lengths. Patch contributed by Devin Cook.

Lax cookie parsing in http.cookies could be a security issue when combined with non-standard cookie handling in some web browsers. Reported by Sergey Bobrov.

logging methods now accept an exception instance as well as a Boolean value or exception tuple. Thanks to Yury Selivanov for the patch.

An exception in Tkinter callback no longer crashes the program when it is run with pythonw.exe.

Prevent turtle AttributeError with non-default Canvas on OS X.

sqlite3 now raises an exception if the request contains a null character instead of truncating it. Based on patch by Victor Stinner.

The glob module now supports recursive search in subdirectories using the `**` pattern.

Fixed a crash in Tkinter on AIX when called Tcl command with empty string or tuple argument.

Tkinter now most likely raises MemoryError instead of crash if the memory allocation fails.

Fix a crash in the json module on memory allocation failure.

imaplib.IMAP4 now supports the context management protocol. Original patch by Tarek Ziadé.

We now override tuple methods in mock.call objects so that they can be used as normal call attributes.

load_tests() is now unconditionally run when it is present in a package's __init__.py. TestLoader.loadTestsFromModule() still accepts use_load_tests, but it is deprecated and ignored. A new keyword-only attribute *pattern* is added and documented. Patch given by Robert Collins, tweaked by Barry Warsaw.

First letter no longer is stripped from the "status" key in the result of Treeview.heading().

Fixed resource leak in the HTTP connection when an invalid response is received. Patch by Martin Panter.

Add a .version() method to SSL sockets exposing the actual protocol version in use.

configparser exceptions no longer expose implementation details. Chained KeyErrors are removed, which leads to cleaner tracebacks. Patch by Claudiu Popa.

turtledemo no longer reloads examples to re-run them. Initialization of variables and gui setup should be done in main(), which is called each time a demo is run, but not on import.

Turtledemo users can change the code font size with a menu selection or control(command) '-' or '+' or control-mousewheel. Original patch by Lita Cho.

The separator between the turtledemo text pane and the drawing canvas can now be grabbed and dragged with a mouse. The code text pane can be widened to easily view or copy the full width of the text. The canvas can be widened on small screens. Original patches by Jan Kanis and Lita Cho.

Turtledemo buttons no longer disappear when the window is shrunk. Original patches by Jan Kanis and Lita Cho.

time.monotonic() is now always available. `threading.Lock.acquire()`, `threading.RLock.acquire()` and socket operations now use a monotonic clock, instead of the system clock, when a timeout is used.

Add a default number of workers to ThreadPoolExecutor equal to 5 times the number of CPUs. Patch by Claudiu Popa.

smtplib now resets its state more completely after a quit. The most obvious consequence of the previous behavior was a STARTTLS failure during a connect/starttls/quit/connect/starttls sequence.

ctypes' BigEndianStructure and LittleEndianStructure now define an empty __slots__ so that subclasses don't always get an instance dict. Patch by Claudiu Popa.

Fix an occasional RuntimeError in threading.Condition.wait() caused by mutation of the waiters queue without holding the lock. Patch by Doug Zongker.

On UNIX, _PyTime_gettimeofday() now uses clock_gettime(CLOCK_REALTIME) if available. As a side effect, Python now depends on the librt library on Solaris and on Linux (only with glibc older than 2.17).

Use e.args to unpack exceptions correctly in distutils.file_util.move_file. Patch by Claudiu Popa.

The webbrowser module now uses subprocess's start_new_session=True rather than a potentially risky preexec_fn=os.setsid call.

signal.set_wakeup_fd(fd) now raises an exception if the file descriptor is in blocking mode.

inspect.stack() now returns a named tuple instead of a tuple. Patch by Daniel Shahaf.

Fixed Tkinter images copying operations in NoDefaultRoot mode.

Add a *globals* argument to timeit functions, in order to override the globals namespace in which the timed code is executed. Patch by Ben Roberts.

Switch urllib.parse to use RFC 3986 semantics for the resolution of relative URLs, rather than RFCs 1808 and 2396. Patch by Demian Brecht.

Added the "members" parameter to TarFile.list().

Allow compileall recursion depth to be specified with a -r option.

Add a __sizeof__ implementation for mmap objects on Windows.

Avoided reference loops with Variables and Fonts in Tkinter.

SimpleHTTPRequestHandler now supports undecodable file names.

Optimized line reading in io.BytesIO.

Raise HTTPError on failed Basic Authentication immediately. Initial patch by Sam Bull.

Restored the use of lazy iterkeys()/itervalues()/iteritems() in the mailbox module.

Changed FeedParser feed() to avoid O(N^2) behavior when parsing long line. Original patch by Raymond Hettinger.

The functools LRU Cache decorator factory now gives an earlier and clearer error message when the user forgets the required parameters.

glob() patterns ending with a slash no longer match non-dirs on AIX. Based on patch by Delhallt.

Added support for RFC 6531 (SMTPUTF8) in smtpd.

Update the ctypes module's libffi to v3.1. This release adds support for the Linux AArch64 and POWERPC ELF ABIv2 little endian architectures.

Added support for the "xztar" format in the shutil module.

Don't force 3rd party C extensions to be built with -Werror=declaration-after-statement.

Fixed crash when using uninitialized sqlite3.Row (in particular when unpickling pickled sqlite3.Row). sqlite3.Row is now initialized in the __new__() method.

Convert posixmodule to use Argument Clinic.

Add an *exists_ok* argument to *Pathlib.mkdir()* to mimic *mkdir -p* and *os.makedirs()* functionality. When true, ignore FileExistsErrors. Patch by Berker Peksag.

Bypass IDNA for pure-ASCII host names in the socket module (in particular for numeric IPs).

set the default value for the *convert_charrefs* argument of HTMLParser to True. Patch by Berker Peksag.

Add an __all__ to html.entities.

the strict mode and argument of HTMLParser, HTMLParser.error, and the HTMLParserError exception have been removed.

Dropped support of Tk 8.3 in Tkinter.

Now Tkinter correctly handles bytes arguments passed to Tk. In particular this allows initializing images from binary data.

When initialized from a bytes object, io.BytesIO() now defers making a copy until it is mutated, improving performance and memory use on some use cases. Patch by David Wilson.

On Windows, signal.set_wakeup_fd() now also supports sockets. A side effect is that Python depends to the WinSock library.

Add os.get_blocking() and os.set_blocking() functions to get and set the blocking mode of a file descriptor (False if the O_NONBLOCK flag is set, True otherwise). These functions are not available on Windows.

Make turtledemo start as active on OS X even when run with subprocess. Patch by Lita Cho.

Fix build error for _multiprocessing when semaphores are not available. Patch by Arfrever Frehtes Taifersar Arahesis.

Convert sha1, sha256, sha512 and md5 to ArgumentClinic. Patch by Vajrasky Kok.

Fix repr(_socket.socket) on Windows 64-bit: don't fail with OverflowError on closed socket. repr(socket.socket) already works fine.

Reprs of most Python implemented classes now contain actual class name instead of hardcoded one.

The dis module can now disassemble generator-iterator objects based on their gi_code attribute. Patch by Clement Rouault.

The asynchat.async_chat.handle_read() method now ignores BlockingIOError exceptions.

Fixed premature DECREF in call_tzinfo_method. Patch by Tom Flanagan.

readline: Disable the meta modifier key if stdout is not a terminal to not write the ANSI sequence `"\033[1034h"` into stdout. This sequence is used on some terminal (ex: TERM=xterm-256color") to enable support of 8 bit characters.

Removed a number of out-of-dated and non-working for a long time Tkinter methods.

Scrollbar.activate() now returns the name of active element if the argument is not specified. Scrollbar.set() now always accepts only 2 arguments.

Clean up and speed up the ntpath module.

plistlib's load() and loads() now work if the fmt parameter is specified.

__qualname__ instead of __name__ is now always used to format fully qualified class names of Python implemented classes.

Reprs now always use hexadecimal format with the "0x" prefix when contain an id in form " at 0x...".

signal.set_wakeup_fd() now raises an OSError instead of a ValueError on `fstat()` failure.

tarfile.open() now handles fileobj with an integer 'name' attribute. Based on patch by Antoine Pietri.

Respect -q command-line option when code module is ran.

Don't pass the redundant 'file' argument to self.error().

Improve exception message of warnings.warn() for bad category. Initial patch by Phil Elson.

os.read() now uses a :c:func:`Py_ssize_t` type instead of :c:type:`int` for the size to support reading more than 2 GB at once. On Windows, the size is truncated to INT_MAX. As any call to os.read(), the OS may read less bytes than the number of requested bytes.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.5.0a1.rst`, line 3037); *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)3.5.0a1.rst`, line 3037); *backlink*

Unknown interpreted text role "c:type".

Fixed source file viewing in pydoc's server mode on Windows.

asynchat.async_chat().set_terminator() now raises a ValueError if the number of received bytes is negative.

asynchat.async_chat.push() now raises a TypeError if it doesn't get a bytes string

Add missing kwonlyargcount argument to ModuleFinder.replace_paths_in_code().

calling Path.with_suffix('') allows removing the suffix again. Patch by July Tikhonov.

Disallow the construction of invalid paths using Path.with_name(). Original patch by Antony Lee.

Added 'auth' method to smtplib to make implementing auth mechanisms simpler, and used it internally in the login method.

Fixed a segfault in the winreg module when `None` is passed as a `REG_BINARY` value to SetValueEx. Patch by John Ehresman.

io.FileIO.readall() does not ignore I/O errors anymore. Before, it ignored I/O errors if at least the first C call read() succeed.

headers parameter of wsgiref.headers.Headers is now optional. Initial patch by Pablo Torres Navarrete and SilentGhost.

ssl.RAND_add() now supports strings longer than 2 GB.

Prevent extraneous fstat() calls during open(). Patch by Bohuslav Kabrda.

cProfile now displays the module name of C extension functions, in addition to their own name.

asyncore: emit a ResourceWarning when an unclosed file_wrapper object is destroyed. The destructor now closes the file if needed. The close() method can now be called twice: the second call does nothing.

Better handling of Python exceptions in the sqlite3 module.

Make sure the email.parser.BytesParser TextIOWrapper is discarded after parsing, so the input file isn't unexpectedly closed.

imghdr now recognizes OpenEXR format images.

Used the "with" statement in the dbm.dumb module to ensure files closing. Patch by Claudiu Popa.

socketserver: Fix a race condition in child processes reaping.

Added the `st_file_attributes` field to os.stat_result on Windows.

Require named tuple inputs to be exact strings.

The distutils "upload" command now exits with a non-zero return code when uploading fails. Patch by Martin Dengler.

asyncio.Queue: support any type of number (ex: float) for the maximum size. Patch written by Vajrasky Kok.

support for "site-python" directories has now been removed from the site module (it was deprecated in 3.4).

new socket.sendfile() method allowing a file to be sent over a socket by using high-performance os.sendfile() on UNIX. Patch by Giampaolo Rodola'.

dbm.dump.open() now always creates a new database when the flag has the value 'n'. Patch by Claudiu Popa.

Add a new is_closed() method to asyncio.BaseEventLoop. run_forever() and run_until_complete() methods of asyncio.BaseEventLoop now raise an exception if the event loop was closed.

Prevent a security hole in CGIHTTPServer by URL unquoting paths before checking for a CGI script at that path.

Fixed possible resource leak in failed open().

Printout of keyword args should be in deterministic order in a mock function call. This will help to write better doctests.

Fixed chaining nonnormalized exceptions in io close() methods.

Fix the pydoc.help function to not fail when sys.stdin is not a valid file.

tempfile.TemporaryFile now uses os.O_TMPFILE flag is available.

Fix pydoc.writedoc so that the HTML documentation for methods that use 'self' in the example code is generated correctly.

In urllib.request, fix pruning of the FTP cache.

The subprocess module could fail to close open fds that were inherited by the calling process and already higher than POSIX resource limits would otherwise allow. On systems with a functioning /proc/self/fd or /dev/fd interface the max is now ignored and all fds are closed.

Introduce importlib.util.module_from_spec() as the preferred way to create a new module.

Fixed possible integer overflow of too long string lengths in the tkinter module on 64-bit platforms.

The zipfile module now ignores extra fields in the central directory that are too short to be parsed instead of letting a struct.unpack error bubble up as this "bad data" appears in many real world zip files in the wild and is ignored by other zip tools.

Added "key" and "reverse" parameters to heapq.merge(). (First draft of patch contributed by Simon Sapin.)

tkinter.ttk now works when default root window is not set.

_tkinter.create() now creates tkapp object with wantobject=1 by default.

sqlite3.Row now truly supports sequence protocol. In particular it supports reverse() and negative indices. Original patch by Claudiu Popa.

If copying (no symlinks) specified for a venv, then the python interpreter aliases (python, python3) are now created by copying rather than symlinking.

Added support for the WebP image type in the imghdr module. Patch by Fabrice Aneche and Claudiu Popa.

Speedup some properties of IP addresses (IPv4Address, IPv6Address) such as .is_private or .is_multicast.

Improve the repr for threading.Lock() and its variants by showing the "locked" or "unlocked" status. Patch by Berker Peksag.

The plistlib module now supports loading of binary plist files when reference or offset size is not a power of two.

Add a default backlog to socket.listen().

Most Tkinter methods which accepted tuples now accept lists too.

With the assistance of a new internal _codecs._forget_codec helping function, test_codecs now clears the encoding caches to avoid the appearance of a reference leak

Tkinter tests now don't reuse default root window. New root window is created for every test class.

Fix PEP 3118 format strings on ctypes objects with a nontrivial shape.

Optimize ipaddress.collapse_addresses().

Optimize ipaddress.summarize_address_range() and ipaddress.{IPv4Network,IPv6Network}.subnets().

Optimize parsing of netmasks in ipaddress.IPv4Network and ipaddress.IPv6Network.

Disallowed the surrogatepass error handler for non UTF-* encodings.

Fixed re.fullmatch() of repeated single character pattern with ignore case. Original patch by Matthew Barnett.

fileinput.FileInput now reads bytes from standard stream if binary mode is specified. Patch by Sam Kimbrel.

Add a samefile() method to pathlib Path objects. Initial patch by Vajrasky Kok.

Set up modules properly in PyImport_ExecCodeModuleObject (and friends).

Fix a unicode error in the pydoc pager when the documentation contains characters not encodable to the stdout encoding.

ipaddress.IPv4Network and ipaddress.IPv6Network now accept an (address, netmask) tuple argument, so as to easily construct network objects from existing addresses.

importlib.abc.InspectLoader.source_to_code() is now a staticmethod.

Simplified and optimized heaqp.nlargest() and nmsmallest() to make fewer tuple comparisons.

Fix TextIOWrapper(..., write_through=True) to not force a flush() on the underlying binary stream. Patch by akira.

Unlink now removes junctions on Windows. Patch by Kim Gräsman

Bugfix for curses.window.addch() regression in 3.4.0. In porting to Argument Clinic, the first two arguments were reversed.

_decimal: The module now supports function signatures.

Remove the non-standard 'watchexp' parameter from the Decimal.quantize() method in the Python version. It had never been present in the C version.

Reduced the risk of false positives in robotparser by checking to make sure that robots.txt has been read or does not exist prior to returning True in can_fetch().

Have the OrderedDict mark deleted links as unusable. This gives an early failure if the link is deleted during iteration.

Add __slots__ to the MappingViews ABC. Patch by Josh Rosenberg.

Eliminate double hashing in the C speed-up code for collections.Counter().

itertools.islice() now releases the reference to the source iterator when the slice is exhausted. Patch by Anton Afanasyev.

TextIOWrapper now allows the underlying binary stream's read() or read1() method to return an arbitrary bytes-like object (such as a memoryview). Patch by Nikolaus Rath.

SSLSocket.send() now raises either SSLWantReadError or SSLWantWriteError on a non-blocking socket if the operation would block. Previously, it would return 0. Patch by Nikolaus Rath.

removed previously deprecated asyncore.dispatcher __getattr__ cheap inheritance hack.

assertRaises now tries to clear references to local variables in the exception's traceback.

ssl.cert_time_to_seconds() now interprets the given time string in the UTC timezone (as specified in RFC 5280), not the local timezone.

Calling sys.flags.__new__ would crash the interpreter, now it raises a TypeError.

Make operations on a closed dbm.dumb database always raise the same exception.

Detect when the os.urandom cached fd has been closed or replaced, and open it anew.

subprocess's Popen.wait() is now thread safe so that multiple threads may be calling wait() or poll() on a Popen instance at the same time without losing the Popen.returncode value.

Path objects can now be instantiated from str subclass instances (such as `numpy.str_`).

urllib.response object to use _TemporaryFileWrapper (and _TemporaryFileCloser) facility. Provides a better way to handle file descriptor close. Patch contributed by Christian Theune.

minidom now raises a custom ValueError indicating it doesn't support spaces in URIs instead of letting a 'split' ValueError bubble up.

The ssl.PROTOCOL* constants are now enum members.

posixmodule: Don't define USE_XATTRS on KFreeBSD and the Hurd.

New method assert_not_called for Mock. It raises AssertionError if the mock has been called.

New keyword argument *unsafe* to Mock. It raises *AttributeError* incase of an attribute startswith assert or assret.

ssl.get_server_certificate() now uses PROTOCOL_SSLv23, not PROTOCOL_SSLv3, for maximum compatibility.

patch.stopall() didn't work deterministically when the same name was patched more than once.

Updated fileConfig and dictConfig to remove inconsistencies. Thanks to Jure Koren for the patch.

Passing name keyword argument to mock.create_autospec now works.

Add lib64 -> lib symlink in venvs on 64-bit non-OS X POSIX.

Some SMTP servers disconnect after certain errors, violating strict RFC conformance. Instead of losing the error code when we issue the subsequent RSET, smtplib now returns the error code and defers raising the SMTPServerDisconnected error until the next command is issued.

setting an iterable side_effect on a mock function created by create_autospec now works. Patch by Kushal Das.

Fix `Host:` header and reconnection when using http.client.HTTPConnection.set_tunnel(). Patch by Nikolaus Rath.

unittest.mock.MagicMock now supports division. Patch by Johannes Baiter.

Fix arbitrary memory access in JSONDecoder.raw_decode with a negative second parameter. Bug reported by Guido Vranken. (See also: CVE-2014-4616)

getpass now handles non-ascii characters that the input stream encoding cannot encode by re-encoding using the replace error handler.

Fixed undocumented filter API of the rot13 codec. Patch by Berker Peksag.

Improved math.factorial error message for large positive inputs and changed exception type (OverflowError -> ValueError) for large negative inputs.

isinstance check relaxed from dict to collections.Mapping.

asyncio.EventLoop.create_unix_server() now raises a ValueError if path and sock are specified at the same time.

Avoid unnecessary normalization of Fractions resulting from power and other operations. Patch by Raymond Hettinger.

Introduce importlib.util.LazyLoader.

signal module constants were turned into enums. Patch by Giampaolo Rodola'.

Improved the repr of Tkinter widgets.

The items, keys, and values views of OrderedDict now support reverse iteration using reversed().

Improved thread-safety in logging cleanup during interpreter shutdown. Thanks to Devin Jeanpierre for the patch.

Fix a leak of file descriptor in :func:`tempfile.NamedTemporaryFile`, close the file descriptor if :func:`io.open` fails

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.5.0a1.rst`, **line 4147**); ***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)3.5.0a1.rst`, **line 4147**); ***backlink***

Unknown interpreted text role "func".

Return None from pkgutil.get_loader() when __spec__ is missing.

Enhance ssl.create_default_context() when used for server side sockets to provide better security by default.

assertRaisesRegex and *assertWarnsRegex* now raise a TypeError if the second argument is not a string or compiled regex.

Replace relative import by absolute import.

Stop wrapping exception when using ThreadPool.

In os.makedirs, do not set the process-wide umask. Note this changes behavior of makedirs when exist_ok=True.

Fix issues found by pyflakes for multiprocessing.

SSL contexts will now automatically select an elliptic curve for ECDH key exchange on OpenSSL 1.0.2 and later, and otherwise default to "prime256v1".

Improve the command-line interface of json.tool.

Enhance default ciphers used by the ssl module to enable better security and prioritize perfect forward secrecy.

Don't assume that __file__ is defined on importlib.__init__.

Ignore __builtins__ in several test_importlib.test_api tests.

xmlrpc.client.ServerProxy is now a context manager.

The formatter module now raises DeprecationWarning instead of PendingDeprecationWarning.

Remove the ability of datetime.time instances to be considered false in boolean contexts.

selectors module now supports /dev/poll on Solaris. Patch by Giampaolo Rodola'.

When the `LC_TYPE` locale is the POSIX locale (`C` locale), :py:data:`sys.stdin` and :py:data:`sys.stdout` are now using the `surrogateescape` error handler, instead of the `strict` error handler.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.5.0a1.rst`, line 4309); *backlink*

Unknown interpreted text role "py:data".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d)3.5.0a1.rst`, line 4309); *backlink*

Unknown interpreted text role "py:data".

Implement incremental decoder for cp65001 code (Windows code page 65001, Microsoft UTF-8).

Delay the initialization of encoding and decoding tables for base32, ascii85 and base85 codecs in the base64 module, and delay the initialization of the unquote_to_bytes() table of the urllib.parse module, to not waste memory if these modules are not used.

Include the broadcast address in the usable hosts for IPv6 in ipaddress.

When an external command (e.g. compiler) fails, distutils now prints out the whole command line (instead of just the command name) if the environment variable DISTUTILS_DEBUG is set.

distutils should not produce unhelpful "error: None" messages anymore. distutils.util.grok_environment_error is kept but doc-deprecated.

Prevent possible gzip "'read' is not defined" NameError. Patch by Claudiu Popa.

`email.message.Message.attach` now returns a more useful error message if `attach` is called on a message for which `is_multipart` is False.

RE pattern methods now accept the string keyword parameters as documented. The pattern and source keyword parameters are left as deprecated aliases.

Fix modulefinder to work with bytecode-only modules.

copy.copy() now doesn't make a copy when the input is a bytes object. Initial patch by Peter Otten.

On AIX, time.mktime() now raises an OverflowError for year outsize range [1902; 2037].

inspect.signature: Use enum for parameter kind constants.

inspect.signature: Make Signature and Parameter picklable.

Add inspect.Signature.from_callable method.

Improve repr of inspect.Signature and inspect.Parameter.

Fix inspect.getcallargs() to raise correct TypeError for missing keyword-only arguments. Patch by Jeremiah Lowin.

Fix inspect.getcallargs() to fail correctly if more than 3 arguments are missing. Patch by Jeremiah Lowin.

Ensure a meaningful exception is raised when attempting to parse more than one XML document per pyexpat xmlparser instance. (Original patches by Hirokazu Yamamoto and Amaury Forgeot d'Arc, with suggested wording by David Gutteridge)

Fix inspect.signature to better support functools.partial. Due to the specifics of functools.partial implementation, positional-or-keyword arguments passed as keyword arguments become keyword-only.

inspect.Signature and inspect.Parameter are now hashable. Thanks to Antony Lee for bug reports and suggestions.

doctest.DocTestSuite returns an empty unittest.TestSuite instead of raising ValueError if it finds no tests

Fix asyncio.tasks.CoroWrapper to workaround a bug in yield-from implementation in CPythons prior to 3.4.1.

asyncio: Add gi_{frame,running,code} properties to CoroWrapper (upstream issue #163).

Avoid exception in _osx_support with non-standard compiler configurations. Patch by John Szakmeister.

Ensure that the turtle window becomes the topmost window when launched on OS X.

Validate that __signature__ is None or an instance of Signature.

Prevent AttributeError in distutils.sysconfig.customize_compiler due to possible uninitialized _config_vars.

Fix http.server to again handle scripts in CGI subdirectories, broken by the fix for security issue #19435. Patch by Zach Byrne.

Fix ffi_prep_args not zero-extending argument values correctly on 64-bit Windows.

Default to TCP_NODELAY=1 upon establishing an HTTPConnection. Removed use of hard-coded MSS as it's an optimization that's no longer needed with Nagle disabled.

Configuration of the max line length for the FormatParagraph extension has been moved from the General tab of the Idle preferences dialog to the FormatParagraph tab of the Config Extensions dialog. Patch by Tal Einat.

Update Idle doc chapter to match current Idle and add new information.

Add Idle extension configuration dialog to Options menu. Changes are written to HOME/.idlerc/config-extensions.cfg. Original patch by Tal Einat.

A module browser (File : Class Browser, Alt+C) requires an editor window with a filename. When Class Browser is requested otherwise, from a shell, output window, or 'Untitled' editor, Idle no longer displays an error box. It now pops up an Open Module box (Alt+M). If a valid name is entered and a module is opened, a corresponding browser is also opened.

Save As to type Python files automatically adds .py to the name you enter (even if your system does not display it). Some systems automatically add .txt when type is Text files.

Code objects are not normally pickled by the pickle module. To match this, they are no longer pickled when running under Idle.

Adjust Editor window title; remove 'Python', move version to end.

Idle debugger breakpoints no longer disappear when inserting or deleting lines.

Turtledemo can now be run from Idle. Currently, the entry is on the Help menu, but it may move to Run. Patch by Ramchandra Apt and Lita Cho.

Add support for non-ascii identifiers to HyperParser.

Add unittest for WidgetRedirector. Initial patch by Saimadhav Heblikar.

Add unittest for SearchDialogBase. Patch by Phil Webster.

Add unittest for ParenMatch. Patch by Saimadhav Heblikar.

add unittest for HyperParser. Original patch by Saimadhav Heblikar.

Add missing upper(lower)case versions of default Windows key bindings for Idle so Caps Lock does not disable them. Patch by Roger Serwy.

Closing a Find-in-files output window while the search is still in progress no longer closes Idle.

Add unittest for textView. Patch by Phil Webster.

Add unittest for AutoExpand. Patch by Saihadhav Heblikar.

Add unittest for AutoComplete. Patch by Phil Webster.

htest.py - Improve framework, complete set of tests. Patches by Saimadhav Heblikar

Add idlelib/idle_test/htest.py with a few sample tests to begin consolidating and improving human-validated tests of Idle. Change other files as needed to work with htest. Running the module as __main__ runs all tests.

Change default paragraph width to 72, the PEP 8 recommendation.

Paragraph reformat test passes after user changes reformat width.

Ensure IDLE menus are customized properly on OS X for non-framework builds and for all variants of Tk.

Rename IDLE "Windows" menu item to "Window". Patch by Al Sweigart.

Use standard PKG_PROG_PKG_CONFIG autoconf macro in the configure script.

Allow the ssl module to be compiled if openssl doesn't support SSL 3.

Drop support of the Borland C compiler to build Python. The distutils module still supports it to build extensions.

Drop support of MS-DOS, especially of the DJGPP compiler (MS-DOS port of GCC).

Check whether self.extensions is empty in setup.py. Patch by Jonathan Hosmer.

Remove incorrect uses of recursive make. Patch by Jonas Wagner.

Define HAVE_ROUND when building with Visual Studio 2013 and above. Patch by Zachary Turner.

the programs that embed the CPython runtime are now in a separate "Programs" directory, rather than being kept in the Modules directory.

"make suspicious", "make linkcheck" and "make doctest" in Doc/ now display special message when and only when there are failures.

The Windows build process no longer attempts to find Perl, instead relying on OpenSSL source being configured and ready to build. The `PCbuild\build_ssl.py` script has been re-written and re-named to `PCbuild\prepare_ssl.py`, and takes care of configuring OpenSSL source for both 32 and 64 bit platforms. OpenSSL sources obtained from svn.python.org will always be pre-configured and ready to build.

Add a build option to enable AddressSanitizer support.

The Windows build process now creates "python.bat" in the root of the source tree, which passes all arguments through to the most recently built interpreter.

Refactor and fix curses configure check to always search in a ncursesw directory.

For BerkeleyDB and Sqlite, only add the found library and include directories if they aren't already being searched. This avoids an explicit runtime library dependency.

Tools/scripts/generate_opcode_h.py automatically regenerates Include/opcode.h from Lib/opcode.py if the latter gets any change.

OS X installer build support for documentation build changes in 3.4.1: assume externally supplied sphinx-build is available in /usr/bin.

Eliminate use of deprecated bundlebuilder in OS X builds.

Incorporated Tcl, Tk, and Tix builds into the Windows build solution.

Fix Modules/Setup *shared* support.

Anticipated fixes to support OS X versions > 10.9.

Prevent possible segfaults and other random failures of python --generate-posix-vars in pybuilddir.txt build target.

Fix library order returned by python-config.

Add library build dir for Python extension cross-builds.

Windows build updated to support VC 14.0 (Visual Studio 2015), which will be used for the official release.

Build _msi.pyd with cabinet.lib instead of fci.lib

Use private version of OpenSSL for OS X 10.5+ installer.

Remove obsolete support for view==NULL in PyBuffer_FillInfo(), bytearray_getbuffer(), bytesiobuf_getbuffer() and array_buffer_getbuf(). All functions now raise BufferError in that case.

PyBuffer_IsContiguous() now implements precise contiguity tests, compatible with NumPy's NPY_RELAXED_STRIDES_CHECKING compilation flag. Previously the function reported false negatives for corner cases.

PyType_Ready() now checks that statically allocated type has no dynamically allocated bases.

Removed non-documented macro PyObject_REPR().

Rename `_Py_char2wchar()` to :c:func:`Py_DecodeLocale`, rename `_Py_wchar2char()` to :c:func:`Py_EncodeLocale`, and document these functions.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\(cpython-main) (Misc) (NEWS.d) 3.5.0a1.rst`, **line 5159);** ***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) 3.5.0a1.rst`, **line 5159);** ***backlink***

Unknown interpreted text role "c:func".

Add new C functions: PyMem_RawCalloc(), PyMem_Calloc(), PyObject_Calloc(), _PyObject_GC_Calloc(). bytes(int) is now using `calloc()` instead of `malloc()` for large objects which is faster and use less memory.

PyImport_ImportFrozenModuleObject() no longer sets __file__ to match what importlib does; this affects _frozen_importlib as well as any module loaded using imp.init_frozen().

Update the codecs module documentation to better cover the distinction between text encodings and other codecs, together with other clarifications. Patch by Martin Panter.

Doc/Makefile now supports `make venv PYTHON=../python` to create a venv for generating the documentation, e.g., `make html PYTHON=venv/bin/python3`.

The documentation of the json module now refers to new JSON RFC 7159 instead of obsoleted RFC 4627.

The binary sequence methods on bytes and bytearray are now documented explicitly, rather than assuming users will be able to derive the expected behaviour from the behaviour of the corresponding str methods.

undocument deprecated asynchat.fifo class.

Expanded functionality of the `Doc/make.bat` script to make it much more comparable to `Doc/Makefile`.

Update the thread_foobar.h template file to include newer threading APIs. Patch by Jack McCracken.

Remove the recommendation for specific CA organizations and to mention the ability to load the OS certificates.

Add missing documentation for PurePath.with_name() and PurePath.with_suffix().

New package installation and distribution guides based on the Python Packaging Authority tools. Existing guides have been retained as legacy links from the distutils docs, as they still contain some required reference material for tool developers that isn't recorded anywhere else.

Document cases where __main__.__spec__ is None.

Add tests for CLI of the calendar module.

Added some additional checks to test_codecs to ensure that statements in the updated documentation remain accurate. Patch by Martin Panter.

All test_re tests now work with unittest test discovery.

Update lib2to3 tests to use unittest test discovery.

Convert test_curses to use unittest.

Skip two tests in test_urllib2net.py if _ssl module not present. Patch by Remi Pointel.

Fix test_pdb to run in refleak mode (-R). Patch by Xavier de Gaye.

test_ctypes has been somewhat cleaned up and simplified; it now uses unittest test discovery to find its tests.

regrtest.py no longer holds a reference to the suite of tests loaded from test modules that don't define test_main().

Assorted cleanups in test_imaplib. Patch by Milan Oberkirch.

Added `load_package_tests` function to test.support and used it to implement/augment test discovery in test_asyncio, test_email, test_importlib, test_json, and test_tools.

Fix test_ssl to accept LibreSSL version strings. Thanks to William Orr.

Converted test_tools from a module to a package containing separate test files for each tested script.

Use modern unittest features in test_argparse. Initial patch by Denver Coneybeare and Radu Voicilas.

Changed HTTP method names in failing tests in test_httpservers so that packet filtering software (specifically Windows Base Filtering Engine) does not interfere with the transaction semantics expected by the tests.

Refactored the ctypes test package to skip tests explicitly rather than silently.

All resources are now allowed when tests are not run by regrtest.py.

Fix pystone micro-benchmark: use floor division instead of true division to benchmark integers instead of floating point numbers. Set pystone version to 1.2. Patch written by Lennart Regebro.

Added tests for Tkinter images.

Added test for ntpath.expanduser(). Original patch by Claudiu Popa.

Added tests for the spwd module. Original patch by Vajrasky Kok.

Added Tkinter tests for Listbox.itemconfigure(), PanedWindow.paneconfigure(), and Menu.entryconfigure().

Fix test_code test when run from the installed location.

Fix distutils tests when run from the installed location.

Consolidated checks for GUI availability. All platforms now at least check whether Tk can be instantiated when the GUI resource is requested.

Fix a socket test on KFreeBSD.

Pass test_site/test_startup_imports when some of the extensions are built as builtins.

Added tests for Tk geometry managers.

Add test case for freeze.

Fix a reference leak in test_tcl.

Move test_namespace_pkgs into test_importlib.

Use test_both() consistently in test_importlib.

Avoid various network test failures due to new redirect of http://www.python.org/ to https://www.python.org: use http://www.example.com instead.

asyncio tests no longer rely on tests.txt file. (Patch by Vajrasky Kok)

Prevent failures of ctypes test_macholib on OS X if a copy of libz exists in $HOME/lib or /usr/local/lib.

Prevent some Tk segfaults on OS X when running gui tests.

Workaround test_logging failure on some OS X 10.6 systems.

Prevent test_ssl failures with large OpenSSL patch level values (like 0.9.8zc).

pydoc now works when the LINES environment variable is set.

Argument Clinic now supports the "type" argument for the int converter. This permits using the int converter with enums and typedefs.

The makelocalealias.py script no longer ignores UTF-8 mapping.

The makelocalealias.py script now can parse the SUPPORTED file from glibc sources and supports command line options for source paths.

Command-line interface of the zipfile module now correctly extracts ZIP files with directory entries. Patch by Ryan Wilson.

For functions using an unsigned integer return converter, Argument Clinic now generates a cast to that type for the comparison to -1 in the generated code. (This suppresses a compilation warning.)

Tools/scripts/diff.py now uses argparse instead of optparse.

Make Tools/scripts/md5sum.py work in Python 3. Patch by Zachary Ware.

Fix Argument Clinic's "--converters" feature.

Add support for `yield from` to 2to3.

Add support for the PEP 465 matrix multiplication operator to 2to3.

Fix module exception list and __file__ handling in freeze. Patch by Meador Inge.

Consider ABI tags in freeze. Patch by Meador Inge.

PYTHONWARNING no longer affects the run_tests.py script. Patch by Arfrever Frehtes Taifersar Arahesis.

Update Windows installer

The bundled version of Tcl/Tk has been updated to 8.6.3. The most visible result of this change is the addition of new native file dialogs when running on Windows Vista or newer. See Tcl/Tk's TIP 432 for more information. Also, this version of Tcl/Tk includes support for Windows 10.

The Windows build scripts now expect external library sources to be in `PCbuild\..\externals` rather than `PCbuild\..\..`.

The Windows build scripts now use a copy of NASM pulled from svn.python.org to build OpenSSL.

Improved the batch scripts provided for building Python.

The bundled version of OpenSSL has been updated to 1.0.1j.

Use versioned labels in the Windows start menu. Patch by Olive Kilburn.

.pyd files with a version and platform tag (for example, ".cp35-win32.pyd") will now be loaded in preference to those without tags.