

SSLObject.version() now correctly returns None when handshake over BIO has not been performed yet.

Upgrade libexpat embedded copy from version 2.2.1 to 2.2.3 to get security fixes.

Fix an assertion failure in *subprocess.Popen()* on Windows, in case the env argument has a bad keys() method. Patch by Oren Milman.

Fix an assertion failure in *PyErr_WriteUnraisable()* in case of an exception with a bad `__module__` attribute. Patch by Oren Milman.

Fix assertion failures in case of a bad warnings.filters or warnings.defaultaction. Patch by Oren Milman.

Raise a TypeError instead of SystemError in case warnings.onceregistry is not a dictionary. Patch by Oren Milman.

Fix several possible instances of undefined behavior due to floating-point demotions.

Location information (lineno and col_offset) in f-strings is now (mostly) correct. This fixes tools like flake8 from showing warnings on the wrong line (typically the first line of the file).

Include sys/sysmacros.h for major(), minor(), and makedev(). GNU C library plans to remove the functions from sys/types.h.

Fix an assertion failure in *zipimport.zipimporter.get_data* on Windows, when the return value of `pathname.replace('/', '\\')` isn't a string. Patch by Oren Milman.

Fix an assertion failure in the write() method of *io.TextIOWrapper*, when the encoder doesn't return a bytes object. Patch by Oren Milman.

Fix a crash in some methods of *io.TextIOWrapper*, when the decoder's state is invalid. Patch by Oren Milman.

print now shows correct usage hint for using Python 2 redirection syntax. Patch by Sanyam Khurana.

Fix a race condition in `importlib._get_module_lock()`.

Fix potential crash during GC caused by `tp_dealloc` which doesn't call `PyObject_GC_UnTrack()`.

Avoid masking original TypeError in call with * unpacking when other arguments are passed.

`str.format_map()` now passes key lookup exceptions through. Previously any exception was replaced with a KeyError exception.

Use `_Py_atomic` API for concurrency-sensitive signal state.

Relative import from unloaded package now reimports the package instead of failing with SystemError. Relative import from non-package now fails with ImportError rather than SystemError.

Improve signal delivery. Avoid using `Py_AddPendingCall` from signal handler, to avoid calling signal-unsafe functions. The tests I'm adding here fail without the rest of the patch, on Linux and OS X. This means our signal delivery logic had defects (some signals could be lost).

Avoid blocking in `pthread_mutex_lock()` when `PyThread_acquire_lock()` is asked not to block.

Make sure the 'Missing parentheses' syntax error message is only applied to SyntaxError, not to subclasses. Patch by Martijn Pieters.

Fixed a race condition when import a submodule from a package.

print now shows expected input in custom error message when used as a Python 2 statement. Patch by Sanyam Khurana.

xml.etree: Fix a crash when a parser is part of a reference cycle.

`typing.get_type_hints` now finds the right globals for classes and modules by default (when no `globals` was specified by the caller).

Speed improvements to the `typing` module. Original PRs by Ivan Levkivskyi and Mitar.

The C accelerator module of ElementTree ignored exceptions raised when looking up TreeBuilder target methods in XMLParser().

`socket.create_connection()` now fixes manually a reference cycle: clear the variable storing the last exception on success.

LoggerAdapter objects can now be nested.

Improves SSL error handling to avoid losing error numbers.

`ssl.SSLContext()` now uses OpenSSL error information when a context cannot be instantiated.

`SSLSocket.sendall()` now uses memoryview to create slices of data. This fixes support for all bytes-like object. It is also more efficient and avoids costly copies.

Fix string concatenation bug in rare error path in the subprocess module

Micro-optimize `:func:'asyncio._get_running_loop'` to become up to 10% faster.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\cpython-main [Misc] [NEWS.d] 3.6.3rc1.rst, line 347); [backlink](#)

Unknown interpreted text role "func".

expat: Update libexpat from 2.2.3 to 2.2.4. Fix copying of partial characters for UTF-8 input (libexpat bug 115): <https://github.com/libexpat/libexpat/issues/115>

Add TLS 1.3 cipher suites and OP_NO_TLSv1_3.

Fix concurrent.futures.thread.ThreadPoolExecutor threads to have a non repr() based thread name by default when no thread_name_prefix is supplied. They will now identify themselves as "ThreadPoolExecutor-y_n".

Fix a segmentation fault in _hashopenssl when standard hash functions such as md5 are not available in the linked OpenSSL library. As in some special FIPS-140 build environments.

The map() and as_completed() iterators in concurrent.futures now avoid keeping a reference to yielded objects.

Fix ctypes producing wrong PEP 3118 type codes for integer types.

The subprocess module now sets the filename when FileNotFoundError is raised on POSIX systems due to the executable or cwd not being found.

concurrent.futures: WorkItem.run() used by ThreadPoolExecutor now breaks a reference cycle between an exception object and the WorkItem object.

xmlrpc.server now explicitly breaks reference cycles when using sys.exc_info() in code handling exceptions.

The ssl and hashlib modules now call OpenSSL_add_all_algorithms_noconf() on OpenSSL < 1.1.0. The function detects CPU features and enables optimizations on some CPU architectures such as POWER8. Patch is based on research from Gustavo Serra Scalet.

Fixed miscellaneous errors in asyncio speedup module.

tk: fix the destroy() method of LabeledScale and OptionMenu classes. Call the parent destroy() method even if the used attribute doesn't exist. The LabeledScale.destroy() method now also explicitly clears label and scale attributes to help the garbage collector to destroy all widgets.

Fix copyreg._slotnames() mangled attribute calculation for classes whose name begins with an underscore. Patch by Shane Harvey.

Fixed a crash when using asyncio and threads.

Fix handling of long oids in ssl. Based on patch by Christian Heimes.

ftplib.FTP.putline() now throws ValueError on commands that contains CR or LF. Patch by Dong-hee Na.

multiprocessing.Queue.get() with a timeout now polls its reader in non-blocking mode if it succeeded to acquire the lock but the acquire took longer than the timeout.

Fix unittest.mock's autospec to not fail on method-bound builtin functions. Patch by Aaron Gallagher.

Fix decrementing a borrowed reference in tracemalloc.

Change ttk.OptionMenu radiobuttons to be unique across instances of OptionMenu.

Fix multiprocessing.Queue.join_thread(): it now waits until the thread completes, even if the thread was started by the same process which created the queue.

Fix segfault in readline when using readline's history-size option. Patch by Nir Soffer.

socket.close() now ignores ECONNRESET error.

Fix out of bounds write in asyncio.CFuture.remove_done_callback().

signal.setitimer() may disable the timer when passed a tiny value. Tiny values (such as 1e-6) are valid non-zero values for setitimer(), which is specified as taking microsecond-resolution intervals. However, on some platform, our conversion routine could convert 1e-6 into a zero interval, therefore disabling the timer instead of (re-)scheduling it.

Fix bug when modifying os.environ while iterating over it

Fix email header value parser dropping folding white space in certain cases.

os.listdir() and os.scandir() now emit bytes names when called with bytes-like argument.

Prohibited the '=' character in environment variable names in os.putenv() and os.spawn*().

Fixed the lgettext() family of functions in the gettext module. They now always return bytes.

Fix incomplete code snippet in the ZeroMQSocketListener and ZeroMQSocketHandler examples and adapt them to Python 3.

Add RFC 7525 and Mozilla server side TLS links to SSL documentation.

Clarify doc on truth value testing. Original patch by Peter Thomassen.

Silence traceback in test_ssl

Remove sha256.tbs-internet.com ssl test

Address ALPN callback changes for OpenSSL 1.1.0f. The latest version behaves like OpenSSL 1.0.2 and no longer aborts handshake.

regtest: Exclude tzdata from regtest --all. When running the test suite using --use=all / -u all, exclude tzdata since it makes

test_datetime too slow (15-20 min on some buildbots) which then times out on some buildbots. Fix also regrtest command line parser to allow passing -u extralargefile to run test_zipfile64.

Fix compile error when compiling --without-threads. Patch by Masayuki Yamamoto.

Adds detection of Visual Studio 2017 to distutils on Windows.

Change to building with MSVC v141 (included with Visual Studio 2017)

os.cpu_count() now returns the correct number of processors on Windows when the number of logical processors is greater than 64.

Add a missing xmlns to python.manifest so that it matches the schema.

IDLE code context -- fix code update and font update timers. Canceling timers prevents a warning message when test_idle completes.

IDLE - Update non-key options in former extension classes. When applying configdialog changes, call .reload for each feature class. Change ParenMatch so updated options affect existing instances attached to existing editor windows.

IDLE - Improve rstrip entry in doc. Strip trailing whitespace strips more than blank spaces. Multiline string literals are not skipped.

IDLE - make tests pass with zzdummy extension disabled by default.

Document how IDLE runs tkinter programs. IDLE calls tcl/tk update in the background in order to make live interaction and experimentation with tkinter applications much easier.

IDLE -- fix tk entry box tests by deleting first. Adding to an int entry is not the same as deleting and inserting because int("") will fail.

Rearrange IDLE configdialog GenPage into Window, Editor, and Help sections.

IDLE - Add docstrings and tests for outwin subclass of editor. Move some data and functions from the class to module level. Patch by Cheryl Sabella.

IDLE - Do not modify tkinter.message in test_configdialog.

Convert IDLE's built-in 'extensions' to regular features. About 10 IDLE features were implemented as supposedly optional extensions. Their different behavior could be confusing or worse for users and not good for maintenance. Hence the conversion. The main difference for users is that user configurable key bindings for builtin features are now handled uniformly. Now, editing a binding in a keyset only affects its value in the keyset. All bindings are defined together in the system-specific default keysets in config-extensions.def. All custom keysets are saved as a whole in config-extension.cfg. All take effect as soon as one clicks Apply or Ok. The affected events are '<<force-open-completions>>', '<<expand-word>>', '<<force-open-calltip>>', '<<flash-paren>>', '<<format-paragraph>>', '<<run-module>>', '<<check-module>>', and '<<zoom-height>>'. Any (global) customizations made before 3.6.3 will not affect their keyset-specific customization after 3.6.3. and vice versa. Initial patch by Charles Wohlganger.

IDLE: Factor HighPage(Frame) class from ConfigDialog. Patch by Cheryl Sabella.

Add tests for configdialog highlight tab. Patch by Cheryl Sabella.

IDLE: Factor KeysPage(Frame) class from ConfigDialog. The slightly modified tests continue to pass. Patch by Cheryl Sabella.

IDLE -- stop leaks in test_configdialog. Initial patch by Victor Stinner.

Add tests for configdialog keys tab. Patch by Cheryl Sabella.

IDLE: Calltips use *inspect.signature* instead of *inspect.getfullargspec*. This improves calltips for builtins converted to use Argument Clinic. Patch by Louie Lu.

IDLE - Add an outline of a TabPage class in configdialog. Update existing classes to match outline. Initial patch by Cheryl Sabella.

Factor GenPage(Frame) class from ConfigDialog. The slightly modified tests continue to pass. Patch by Cheryl Sabella.

IDLE - Factor FontPage(Frame) class from ConfigDialog. Slightly modified tests continue to pass. Fix General tests. Patch mostly by Cheryl Sabella.

IDLE - Use ttk widgets in ConfigDialog. Patches by Terry Jan Reedy and Cheryl Sabella.

IDLE - Finish rearranging methods of ConfigDialog Grouping methods pertaining to each tab and the buttons will aid writing tests and improving the tabs and will enable splitting the groups into classes.

IDLE -- Factor a VarTrace class out of ConfigDialog. Instance tracers manages pairs consisting of a tk variable and a callback function. When tracing is turned on, setting the variable calls the function. Test coverage for the new class is 100%.

IDLE: Add more tests for General tab.

IDLE - Improve configdialog font page and tests. In configdialog: Document causal pathways in create_font_tab docstring. Simplify some attribute names. Move set_samples calls to var_changed_font (idea from Cheryl Sabella). Move related functions to positions after the create widgets function. In test_configdialog: Fix test_font_set so not order dependent. Fix renamed test_indent_scale so it tests the widget. Adjust tests for movement of set_samples call. Add tests for load functions. Put all font tests in one class and tab indent tests in another. Except for two lines, these tests completely cover the related functions.

IDLE -- Add more configdialog font page tests.

IDLE: replace 'colour' with 'color' in configdialog.

Add tests for `idlelib.config.IdleConf`. Increase coverage from 46% to 96%. Patch by Louie Lu.

Document coverage details for `idlelib` tests. Add section to `idlelib/idle-test/README.txt`. Include check that branches are taken both ways. Exclude IDLE-specific code that does not run during unit tests.

IDLE: Document `ConfigDialog` tk Vars, methods, and widgets in docstrings. This will facilitate improving the dialog and splitting up the class. Original patch by Cheryl Sabella.

IDLE: Add tests for `ConfigParser` subclasses in `config`. Patch by Louie Lu.

IDLE: Add docstrings to `browser.py`. Patch by Cheryl Sabella.

IDLE: Remove unused variables in `configdialog`. One is a duplicate, one is set but cannot be altered by users. Patch by Cheryl Sabella.

IDLE: In Settings dialog, select font with Up, Down keys as well as mouse. Initial patch by Louie Lu.

IDLE: call `config.IdleConf.GetUserCfgDir` only once.

IDLE: Factor `ConfigChanges` class from `configdialog`, put in `config`; test. * In `config`, put dump test code in a function; run it and unittest in 'if __name__ == '__main__':'. * Add class `config.ConfigChanges` based on `changes_class_v4.py` on bpo issue. * Add class `test_config.ChangesTest`, partly using `configdialog_tests_v1.py`. * Revise `configdialog` to use `ConfigChanges`; see tracker msg297804. * Revise `test_configdialog` to match `configdialog` changes. * Remove `configdialog` functions unused or moved to `ConfigChanges`. Cheryl Sabella contributed parts of the patch.

IDLE: `configdialog` - Add docstrings and fix comments. Patch by Cheryl Sabella.

IDLE: Improve `textview` with docstrings, PEP8 names, and more tests. Patch by Cheryl Sabella.

IDLE: Make several improvements to `parenmatch`. Add 'parens' style to highlight both opener and closer. Make 'default' style, which is not default, a synonym for 'opener'. Make time-delay work the same with all styles. Add help for `config` dialog extensions tab, including help for `parenmatch`. Add new tests. Original patch by Charles Wohlganger.

IDLE: add docstrings to `grep` module. Patch by Cheryl Sabella

IDLE's basic custom key entry dialog now detects duplicates properly. Original patch by Saimadhav Heblikar.

IDLE no longer deletes a character after commenting out a region by a key shortcut. Add `return 'break'` for this and other potential conflicts between IDLE and default key bindings.

Review and change `idlelib.configdialog` names. Lowercase method and attribute names. Replace 'colour' with 'color', expand overly cryptic names, delete unneeded underscores. Replace `import *` with specific imports. Patches by Cheryl Sabella.

IDLE: Verify user-entered key sequences by trying to bind them with tk. Add tests for all 3 validation functions. Original patch by G Polo. Tests added by Cheryl Sabella.

`gdb` integration commands (`py-bt`, etc.) work on optimized shared builds now, too. [PEP 523](#) introduced `_PyEval_EvalFrameDefault` which inlines `PyEval_EvalFrameEx` on non-debug shared builds. This broke the ability to use `py-bt`, `py-up`, and a few other Python-specific `gdb` integrations. The problem is fixed by only looking for `_PyEval_EvalFrameDefault` frames in `python-gdb.py`. Original patch by Bruno "Polaco" Penteado.