



Release Notes

Thespian Python Actor System

By: Kevin Quick <quick@sparq.org>

2020 Nov 29 (#3.10.2)

Thespian Project

TheDoc-06

PUBLIC DOCUMENT



Contents

Thespian uses Semantic Versioning: "major.minor.bugfix", where major version changes indicate incompatible changes, minor versions indicate added functionality that is backwards compatible, and bugfix changes do not change the API.

Summary

2020

Release	Date	Significant Bugfixes	Significant Features
3.10.2	2020-11-29	Fix Issue #69: Actor.logger() Initialize Actor log forwarding level Fix for Python 3.8 "is" literals	
3.10.1	2020-07-14	Fix Issue #63: invalid socket fd Unset THESPLOG_THRESHOLD fix	RunCommand buffer limits removed
3.10.0	2020-03-10	transient_idle wakeups	Adjustable Troupe limits, THESPIAN_BASE_IPADDR

2019

Release	Date	Significant Bugfixes	Significant Features
3.9.11	2019-09-10	globalName precedence for createActor	
3.9.10	2019-08-30	runcommand shutdown, drain, and output fixes	
3.9.9	2019-04-08	Fixed message duplication in troupe	Round-robin use of troupe workers
3.9.8	2019-03-04	Transmit queueing/overrun fixes (Issue #38)	Troupe worker exit delays
3.9.7	2019-02-11	Further bugfixes and efficiencies	
3.9.6	2019-02-09	Improve Issue #35: Windows issues Improve Issue #38: system limits	

2018

Release	Date	Significant Bugfixes	Significant Features
3.9.5	2018-12-17	Fix Issue #32: Deadlock in TX backoff	
3.9.4	2018-08-05	Fix Issue #28: Enable offline	
3.9.3	2018-07-31	Log deserialization errors at ERROR level More thread-safety fixes Fix Issue #27: parent dies before child starts Updates to the http_server.	THESPLOG_THRESHOLD
3.9.2	2018-01-31	RunCommand error processing fix	
3.9.1	2018-01-09	RunCommand logging fix	log message serialization errors

2017

Release	Date	Significant Bugfixes	Significant Features
3.9.0	2017-12-22	Issue #9	<code>self.globalName</code> , <code>runcommand</code> helper
3.8.3	2017-11-08	<code>globalName</code> w/convention	
3.8.2	2017-11-01	<code>inspect</code> deprecation, <code>message str()</code> exceptions	
3.8.1	2017-10-22	starting loaded source Actors under Windows thread-protection for <code>wakeupAfter()</code>	enhance examples for Windows
3.8.0	2017-08-22	Issue #5 thread protection	<code>Actor.__init__(capabilities)</code> <code>self.wakeupAfter()</code> payload
3.7.3	2017-05-02	TCP transport trailing input <code>MultiprocQueueBase</code> daemonized background	
3.7.2	2017-03-29	fixed <code>setup.py</code> stale references	
3.7.1	2017-03-29	Director response detection, admin startup socket closure	
3.7.0	2017-03-19	<code>ActorSystem</code> shutdown, transmit completions	<code>ActorSystem.private()</code> context
3.6.3	2017-03-02	revert 3.6.2 regression	
3.6.2	2017-02-19	convention notifications, shutdown cleanup, address comparison commutativity	
3.6.1	2017-01-18	Director locations (support Debian/Ubuntu), Director Python2.6 + Mac <code>launchd</code> fix, Issue 32, message stuck on tx queue	
3.6.0	2017-01-02	Actor Address display, dead target forwarding, preserve loaded source information	Thespian Director, <code>@transient</code> , timer improvements, Source Authority active rejection, address re-use.

2016

Release	Date	Significant Bugfixes	Significant Features
3.5.2	2016-12-12	Slower hysteresis backoff, async tx mgmt fixes with spin/promela models	
3.5.1	2016-12-08	Rollback admin addr optimization, Prioritize local Admin traffic	
3.5.0	2016-12-04	Close idle TCP sockets when needed and add protection against remotely closed sockets, more socket closure when creating children, set TZ in env (avoid stat() to /etc/localtime), troupe worker exit handling Windows SO_EXCLUSIVEADDRUSE for probing running	Actor signal handling, troupe leader status reporting, <code>self.actorSystemShutdown()</code> , Address string brevity
3.4.0	2016-11-07	TCP transport socket shutdown corner cases, logging error (Issue 23)	<code>self.notifyOnSourceAvailability()</code> , <code>ask()</code> and <code>listen()</code> speed, convention management stability, <code>PoisonMessage.details</code> , <code>@troupe</code>
3.3.0	2016-09-15	TCP retransmits on corrupted ack/nak, internal non-progress <code>select()</code> CPU consumption	<code>deRegisterREmoteSystem(remoteAdmin)</code> , Python 3.6 support, convention mgmt,
3.2.2	2016-09-15	No pre-registered on remote notifications, maximize async sends, fix race condition on multiple <code>globalName</code> creates	
3.2.1	2016-09-08	Add backward compatibility re: <code>sourceInfo</code>	
3.2.0	2016-09-08		Added <code>sourceInfo</code> to <code>ValidateSource</code> , multi-threaded Actor <code>self.send()</code>
3.1.1	2016-09-08	Missing import on exception, better TCP connects, disable ActorAddress hashing	
3.1.0	2016-08-29	More complete Actor shutdowns, TCP connect timeout to support busy networks, no infinite-repeat of convention source unload notifications	"Process Startup Method" capability, thespian shell Python3 support, initial multi-threaded Actor support
3.0.4	2016-08-12	Fix init hardening, longer Actor startup allowance	Optimize <code>__import__</code> interceptor.
3.0.3	2016-08-11	Hardened init for fqdn, hostname, or dns gaierrors. Error reporting for actor creation exceptions.	
3.0.2	2016-07-29	Defer unknown createActor sourceHash to Admin SourceLoad import hooking updates.	
3.0.1	2016-07-26	Fix unloads for None sys.modules entries, sourceload import optimizations, more pending actor timeout info	
3.0.0		No hash attempts for ActorAddress socket accept failure handling, pre-reg fixes, sourceload fixes, TCP transport improvements, revert signal handling from 2.5.8, convention handler standardization.	Use <code>py.test</code> for testing
TheDog-96 2.6.1 PUBLIC DOMAIN	2016-05-20	Fix DeadEnvelope in simpleSystemBase Fix SIGUSR1 for Window {^}	2020 Nov 29 Rev. 3.10.2 4 of 29
2.6.0	2016-05-18	No restart of logger on shutdown	SIGUSR1 status reporting, simpleSystemBase capability updates

3.x

3.10

3.10.2 <2020-11-29 Sun>

- Fixed Issue #69: ensure the Actor self.logger() works for all system bases. Previously it only worked properly for the simpleSystemBase.
- Updated Actor initialization to only forward logging messages to the Logger Actor if the message level is above the minimum level in the supplied logDefs at ActorSystem initialization time. This helps prevent transmit overruns when performing significant amounts of ignored logging (e.g. DEBUG level).
- Fix to use "==" instead of "is" for comparing literals (thanks to Peter Sabaini).

3.10.1 <2020-07-14 Tue>

- Fixed Issue #63: properly handle a socket fd that is no longer valid.
- Fixed setting logging threshold level when THESPLOG_THRESHOLD is not set.
- Removed incoming buffer limits for RunCommand, improving performance for running commands with significant amounts of output.
- Misc typing comments and initial mypy.ini file.

3.10.0 <2020-03-10 Tue>

- Added ability to dynamically update the Thespian Troupe worker counts (see the Actor Troupe Support section).
- Added ability to specify the local IP address to use for TCP and UDP actor bases via the THESPIAN_BASE_IPADDR environment variable.
- Fixed bug in the wakeup scheduling for the transient_idle helper.
- Updated the introductory documentation to describe using Thespian for distributed application configurations.
- Clarify the TLI file contents and usage in the Director documentation.
- Other documentation fixes and updates.

3.9

3.9.11 <2019-09-10 Tue>

- Fixed `createActor()` calls so that a `globalName` match is performed before the actor class lookup. This is particularly important for loaded sources where the external segment does not need to have the loaded source definitions when invoking a `globalName` entry to the loaded source.

This also consolidated code for actor creation for single implementation of logic and conformity across endpoints.

- Updated the `rsasig.py` (used for Thespian Director signing/verification) to use the new `tobytes()` instead of the deprecated `tostring()` method on array types.
- Updated Thespian Director list output to more clearly indicate when a tli-driven actor startup has failed.
- Added some additional timeouts to `ask()` calls in the source loading tests.

3.9.10 <2019-08-30 Fri>

- Fixed syntax error handling of large output in `runcommand` that would cause a crash when copious amounts of output were generated by the command that was run.
- Fixed handling of `runcommand` subprocess shutdown for conditions where the subprocess is already exited and the file descriptors are closed.
- Ensure that the `runcommand` subprocess `stdout` and `stderr` descriptors are fully drained before returning.
- Update `py.test run_unstable_tests` fixture to remove deprecated usage.
- Ensure thespian troupe work completion removal doesn't try to remove already removed items.

3.9.9 <2019-04-08 Mon>

- Fixed issue in the troupe where a single message might get delivered to multiple troupe members for handling. This error was revealed during handling of Issue #43 (thanks, andatt!).
- Updated the troupe worker usage to be round-robin, spreading the work more equably across troupe members instead of using a LIFO strategy which focused the work on a single member.

3.9.8 <2019-03-05 Tue>

- Fixes for transmit handling to prevent excessive recursion. This improves the backoff and retry logic in cases where a large number of messages is sent more quickly than can actually be sent, allowing them to be eventually sent.
- Added stats for number of actor create requests and completions.

- Enhance timer drain message with time limit.
- Update to allow troupe leader to receive ActorSystemMessages.
- Allow troupe workers to remain present for a short period of time after completing any current work to reduce the churn on creation/deletion of troupe workers.
- Catch exceptions during child actor creation and handle as ChildActorExited events.
- Many of the above fixes were as a result of extended testing and patience from Andrew (andatt) via Issue #38... thanks, Andrew!
- Miscellaneous test warning cleanups.

3.9.7 <2019-02-11 Mon>

- Remove extraneous messages to ActorSystem at Actor shutdown time.
- Additional shutdown activity suppression on failed startup.
- Update multiprocUDPBase handling of signal interruptions.

3.9.6 <2019-02-05 Tue>

- Improved Issue #35 which addressed some issues running under Windows and also with the multiprocUDPBase. The improvements include PullReq #36 from TTimo (thanks!).
- Improved Issue #38 which addresses the behaviour of Thespian when system limits (e.g. process count or file-descriptor count) are reached.

3.9.5 <2018-12-17 Mon>

- Fixed Issue #32 which caused an actor to hang due to an internal deadlock if the number of queued messages reached the backoff threshold.

3.9.4 <2018-08-05 Sun>

- Fixed Issue #28 which prevented using the multiprocTCPBase or multiprocUDPBase ActorSystem bases offline (i.e. when internet connectivity was unavailable).

3.9.3 <2018-07-31 Tue>

- Added the THESPLUG_THRESHOLD environment variable to allow control over thesplog logging levels without editing files, and fixed some log deserialization (thanks to Daniel Mitterdorfer for this enhancement).

- Set the level for logging incoming message deserialization levels to ERROR (again, thank you to Daniel Mitterdorfer).
- Fixed Issue #27 which prevented a child from being destroyed when the parent exited before the child was successfully created.
- More fixes for thread-safety when Actors use multiple threads.
- Updates to the `http_server` example for handling: remote closure after send, retrieving the full Content-Length specified, and providing the body of the data in the `HTTPRequest` message.
- Internal fixes to `async` transmit to exit when no more work is available.
- Added `test_bigmessages` to test the transfer of very large messages.
- Increased incoming message buffer size for less re-assembly.

3.9.2 <2018-01-31 Wed>

- Fixed an error in the error handling code for the `RunCommand` helper.

3.9.1 <2018-01-09 Tue>

- Message serialization failures (causing a `PoisonMessage` response) are now logged for easier diagnosis.
- Fixed an error in logging for the new `RunCommand` helper.
- Updated `RunCommand` to take a logger Actor Address instead of attempting to call a logging function directly. This allows any filtering and log adjustments to be implemented by the logging actor and keeps `RunCommand` cleaner.

3.9.0 <2017-12-22 Fri>

- Added `self.globalName` to Actor objects. An actor can determine its `globalName` by checking this property: the value will be an empty string if this Actor does not have a `globalName`. Thanks to Waqar Aqeel for this contribution.
- No longer use deprecated `imp` module for newer Python versions. In conjunction with contributions to 3.8.2 from asevas, this fixes Issue #9.
- Logging levels have been adjusted to generate less logging for normal operations. Additionally lowered the default log reporting level for the `simpleSystemBase` from `WARNING` to `INFO` to support its use as a diagnostic/debugging base.
- Changes to the Thespian Director:
 - Added documentation for the Thespian Director, which was introduced in version 3.6.0.
 - The `gensrc` command for the Thespian Director supports recursive globbing for Python versions that support it (see the `glob` module documentation) and has additional output when `verbose` mode is used.

- The "start on load" functionality of the Thespian Director now records (and reports) the start time of the Actor. In addition, there is a minimum running time that a started actor must achieve before it will be restarted on failure; this helps prevent an endless loop of attempts to start one of these Actors cannot properly start.
- The Director is updated to use the installed TLI file instead of the local TLI file for implementing the TLS_Keep_Limit functionality.
- Added the `thespian.runcommand` helper module which provides an Actor and associated messages that can be used for running a command in a sub-process (similar to the Python `subprocess` module) using Thespian best practices to maintain a responsive actor and handle input, output, and result messaging.
- Added a `README.org` file in the thespian directory to describe the (growing) contents.
- Miscellaneous test and documentation updates.

3.8

3.8.3 <2017-11-08 Wed>

- Fixed issue preventing use of `globalName` actors in conjunction with a multiple ActorSystem Convention.
- Fixed documentation link in the Using Thespian documentation.
- Updated description of the `globalName` argument in the Using Thespian documentation.
- Added a better description of the startup modes for an ActorSystem relating to the arguments supplied.

3.8.2 <2017-11-01 Wed>

- Fixed deprecation warnings related to `inspect` module (with thanks to Andrew Evans for the patch).
- Add unit tests and protection against messages that throw exceptions on `str()` translations (with thanks to Felipe Gutierrez for discovering this).
- Fix exception retry handling for `simpleSystemBase` (with thanks to Felipe Gutierrez for discovering this).
- Internal change to child actor startup to remove potential race condition on startup management.
- Added missing `UpdateWork` import for internal Admin functionality.
- Remove support claim for Python 2.6.

3.8.1 <2017-10-22 Sun>

- Fix startup of loaded-source Actors on Windows: use string classname since Windows spawn does not inherit modified `sys.modules` at the time that the startup arguments (which include the child class) are deserialized.

- Update httpserver to use `msvcrt.getch()` instead of `signal.pause()` under Windows (thanks to Thomas Christian for this contribution).
- Update `multi_system` act examples to add `main` wrappers to primary scripts to improve Windows support (thanks to Thomas Christian for this contribution).
- Adds thread synchronization protection to `wakeupAfter()` Actor method in case the Actor is multi-threaded.
- Internal modification of time sampling to allow consistent sampling for logic evaluation.

3.8.0 <2017-08-22 Tue>

- The Actor `__init__()` can now specify capabilities and requirements arguments.

```
class MyActor(Actor):
    def __init__(self, requirements, capabilities):
        if requirements['key'] == 'value' and \
            capabilities['key2'] == 'another value':
            self.mode = 'enabled'
```

If specified, these will receive the corresponding ActorSystem values at the time of the creation of that Actor.

- Added an optional payload to `self.wakeupAfter()` calls. The payload will be returned to the Actor in the subsequent `WakeupMessage` and can be used for whatever the Actor needs, including to differentiate between different wakeups.

Thanks to Daniel Mitterdorfer for contributing this enhancement!

- Update `@requireCapability` decorator to look for truthy values when no explicit value is supplied. Fixes Issue #5.
- Fixed race conditions for situations where multi-threaded actors could encounter an internal Thespian error when multiple threads were sending and the main thread was receiving at the same time.
- Added `pytest` dependency for `tox` specification (thanks to Daniel Mitterdorfer).
- The `tox`-based testing will now test the TCP transports as well.

3.7

3.7.3 <2017-05-02 Tue>

- Protection against malformed trailing input for TCP transport.
- Properly daemonize background thread used for `MultiprocQueueBase` to allow normal shutdown and cleanup.
- Enabled Travis CI testing (thanks to David Asabina!).
- Updates for shift of development to <https://github.com/kquick/Thespian>.
- Miscellaneous test updates to address test-centric instabilities, including avoiding the use of ephemeral ports for test admin ports to avoid collisions, auto-kill of orphaned actors, and renaming test actors for distinction.

3.7.2 <2017-03-29 Wed>

- Removed setup.py and doc references to previously removed thespianShell, which is replaced by `python -m thespian.shell`.

3.7.1 <2017-03-29 Wed>

- Better internal detection of director responses to avoid exceptions on unexpected messages.
- Hold messages sent to Admin until Admin is fully initialized.
- Close **all** sockets to a lost remote system.

3.7.0 <2017-03-19 Sun>

- The major enhancement in this release is the `ActorSystem.private()` method, which can be used by external applications to get a unique endpoint for communicating with actors. This is especially helpful for multi-threaded external applications (Issue 35).

Note that this is not fully supported for the `MultiprocQueueBase` at this time. The `MultiprocQueueBase` usually works, but there are unresolved corner cases at this time.

See the description of the `private` method in the Using Thespian document for more details.

- Fixes a bug where internal transmit events were not completed properly and blocked subsequent transmits.
- Cleaner `ActorSystem` shutdown, driven by responses rather than timed, recursive transport runs.
- Fixed amount of time reported in `ActorSystem.tell()` timeouts.
- Retry TCP transmits immediately on slot availability instead of waiting a delay period.
- Fixed bugs in `ReceiveBuffer` when additional unexpected data is received.
- Consolidated on using `ExpirationTimer` and removed `ExpiryTime`.
- Fixed statistics report of any transmits pending address resolution.
- Write `logdirector` exceptions to `thesp` and fix `logdirector`'s exception recurrence detection.
- Ensure convention registration timer is always refreshed if expired.
- Upgrade UDP transport `select()` error log message to critical level.
- Internally updated transport run method return values for more explicit exit information.
- Internal refactoring, cleanup, and updates.
- Stable documentation hyperlink targets (based on Lee Hinman's article).

3.6

3.6.3 <2017-03-02 Thu>

- Revert changes causing a regression in 3.6.2 wherein messages are dropped or not delivered (Issue 37).

3.6.2 <2017-02-19 Sun>

- Use cPickle when running under older python versions for improved performance (Issue 34).
- Update director bootstart for systemd to add "Want network-online.target" as well as "After network-online.target" to express dependency as well as ordering.
- Ensure address equality comparisons are commutative.
- Explicitly close transports on shutdown to avoid resource cleanup issues.
- Fixed bug where convention notifications for remotes that had only pre-registered could be delivered upon registration.
- Refactor external interface to use individual operation tracking instead of globals.
- Improved detection of shutdown and associated cleanup status and completion.
- Fix actor failure tests for more intentional testing of various actor failure modes.
- Misc internal cleanups to Queue transport; fixes for timeouts and dead target handling.
- Misc internal cleanups in UDP transport; fix handling of child management interrupts and signals from exited children.
- Misc test cleanups.

3.6.1 <2017-01-18 Wed>

- Updated Director to not require root permissions for writing log files when no log files need to be written.
- Update Director bootstart location for systemd to use a more common location (adds support for Debian/Ubuntu).
- Fix Director Python2.6 and Mac launchd compatibility.
- Allow specification of format encoding for Director gensrc to override default.
- Added the ActorAddressLogFilter to the Director to allow logging format specifications to show the actorAddress field even if it is not originally present in the log message.
- Fix from kwadl for ActorSystem initialization under Windows with Python 2.7 (Issue 32).
- Fix for arguments on logger restart conditions.
- Make inter-convention hysteresis more careful about detecting duplicate packets.

- Logging updates, including reduction of severity to improve signal-to-noise ratio for actual errors.
- Show local instance number for ActorAddresses even when actual translation is shown.
- Fix for transmit queue to ensure messages are not stuck on the queue if address updates are performed post-queueing.
- Update setup in tests to ensure proper socket closure for re-use.
- Improved status output for pending transmits.

3.6.0 <2017-01-02 Mon>

- Enhancements
 - Added the thespian director to help manage loading and managing sources. See the `thespian/director.py` file for more information, or `$ python -m thespian.director` for cli help.
 - Added `thespian.transient` import with `@transient` and `@transient_idle` decorators. See `thespian/transient.py` for more information.
 - Added the `rsasig.py` helper source (used by the director) to perform signature validation of sources on systems where a full encryption/SSL library may not be installed or available.
 - Internal improvements to the `ExpirationTimer` and `ExpiryTimer` implementation to reduce overhead for better performance (thanks to Daniel Mitterdorfer).
 - A Source Authority can now actively reject a loaded source by responding with a `None` value for the validated results. Previously the Source Authority could only passively reject the source by doing nothing, which did not provide feedback to waiting elements about the disposition of the load attempt.
 - When displaying Actor Addresses, the final value is shown (if known); previously these always output as "LocalAddr" even if the final address was known.
 - Added a failure reason to the `InvalidActorSpecification` exception.
 - More logging on Actor System startup and shutdown.
 - Transports that may re-use addresses (e.g. TCP ports) do not need to declare an Actor Address as permanently dead-lettered in case of re-use.
- Bugs
 - Ensure that loaded source information specified by an Actor is preserved.
 - Fixed a bug introduced in version 3.5.0 where the string form of Actor Addresses for remote actors might not show the remote address portion.
 - Ensure dead target transmit failures are forwarded to the dead letter handler.
 - Clarified that only `.py` files in the loadable sources are used, and that `.pyo` and `.pyc` files are ignored. This has always been the way it operated, but the previous documentation stated the inverse.
- Other
 - Miscellaneous internal cleanup.
 - Update status reporting of pending wakeups.
 - Refactor and improve `simpleSystemBase` handling of wakeups.

3.5

3.5.2 <2016-12-12 Mon>

- Slow down hysteresis backoff rate for inter-system admin messages.
- Updates to asynchronous transmit management for better multi-threading support, and addition of Spin/Promela models for validation of this behavior.

3.5.1 <2016-12-08 Thu>

- Rollback of Admin routing address creation optimization: loses important information needed when passing the address to remote convention members.
- Ensure local Admin always receives packets, even under high backlog conditions where normal actors will enter TX only mode to drain the outbound message queues.

3.5.0 <2016-12-04 Sun>

- Enhancements
 - Signal handling is re-enabled after being disabled in 3.0.0 with a new approach to avoid conflicts with signal processing context.
 - Only passes class name and not entire class module heirarchy to setproctitle for setting the Actor process name.
 - Added ability to query the troupe leader for status (intended for user interpretation with thespianShell and not programmatic consumption).
 - Added `self.actorSystemShutdown()` to allow an actor to shutdown the entire local actor system (Issue 24).
- Bug Fixes and internal changes
 - The TCPTransport will attempt to close old idle sockets if it fails to create new sockets (incoming or outbound) due to resource limits.
 - Added exception protection against closed descriptors in TCPTransport idle socket cleanup.
 - Now closes all open idle sockets when spawning a child process with the TCPTransport instead of just a subset.
 - Remove child liveness detection confirmation delay (enabled SIGCHLD handling makes this unnecessary).
 - Ensure convention Hysteresis is reset on pre-registration of remote and slow down hysteresis rate.
 - Added status indication of exiting condition in status reports.
 - Internal cleanup of routed address definitions.

- Updated visual form of addresses for more brevity. Note that all code should treat Actor Addresses as opaque, and there should be no programmatic dependence on a specific representation, so this update does not constitute a compatibility change relative to previous versions.
- Added optimization of AdminRouting: local targets are sent directly instead of routing via the Admin, and also avoid extra hops in routing from the Admin back to itself.
- Upgraded logging of transmit completion errors to ERROR from INFO.
- Set the TZ environment variable on startup if not already set to avoid repeated stat() calls to "/etc/localtime" under Unix.
- Fixes and updates to the troupe implementation to handle ChildActorExit messages from workers, ensure completed work messages are sent back to the troupe leader, and use string class names for compatibility with loaded sources.
- Use SO_EXCLUSIVEADDRUSE on Windows to probe for existing multiprocTCPBase.

3.4

3.4.0 <2016-11-07 Mon>

- Added notifyOnSourceAvailability() call for an Actor to register to receive LoadedSource objects indicating that a new source file has been loaded and is ready to be used, and UnloadedSource objects when a source has been unloaded. This can be used to write a "source manager" that can take specific actions when these trigger messages indicate changes in the loaded sources for the current actor system.
- Updated Actor System ask() and listen() calls to return as soon as a response is available. Previously the calls did not return until the delay period had expired.
- Updated convention management with a full analysis of different scenarios (in the design-routing document) and corresponding tests, with updates to the convention management code as revealed by these tests. This fixes various behaviors, including an issue introduced in the previous release that would prevent a TxOnly admin from updating remote nodes and therefore causing the remote to enter a partially-disconnected state after about 22 minutes. These changes introduce significant stability and behavior implements for conventions.
- Add .details to PoisonMessage for additional information (Issue 22).
- Adds supported Actor Troupe functionality (Issue 2).
- Fixed logging error (Issue 23) introduced in 3.3.0.
- Updated packaging to include documentation and top-level files including the readme, license, and contributor's agreement (Issue 25).
- Internal cleanups and re-arrangement, including introduction of a more efficient internal timer.
- Update TCP transport to use more efficient timer, handle socket shutdown corner cases (e.g. no socket created yet or already removed).

3.3

3.3.0 <2016-09-15 Thu>

- Allow `deRegisterRemoteSystem()` to be called with the remote admin `ActorAddress` object (as would be provided via convention notification) as an alternative to the raw remote address format.
- Enhanced the `thespianShell "ask"` command to pretty-print the responses (if possible).
- Thespian is now compatible with Python 3.6
 - Removed the "universal newline" loaded source open flag that is no longer supported for `ZipFile.open()` in Python 3.6.
 - Updated tests to account for address information now included in default `str` forms of objects.
- Internal improvements in convention support for more reliable convention membership and notification.
 - When a remote pre-registration is performed, there is no notification until the convention attendee completes the checkin.
 - Better matching of loopback addresses with the "external" address(es) of the current system to recognize this system.
 - Better connection establishment for pre-registered remote convention members with local tx-only admin routing.
 - Allow more communication with remote convention members on shutdown for a faster and more complete cleanup and exit.
- Internal improvements and bugfixes in the TCP transmit.
 - Enable TCP transport retransmits if ack/nak packet is corrupted.
 - Avoid non-progress `select()` scenarios that could result in periods of high CPU utilization.
 - Better handling of cleanup operations when shutting down.
- Fixed handling for a race condition on timeout in the UDP transport.

3.2

3.2.2 <2016-09-15 Thu>

- Suppressed notification of pre-registered remotes on `preRegisterRemoteSystem()` call; notification is now only delivered when the remote actually confirms convention registration.
- Fixed convention notification remote admin address value (in some scenarios this delivered the convention message itself rather than the remote address).
- Enhanced notification tests.

- Actors with multiple threads that initiate large numbers of sends from the alternate threads but which do not receive many messages to invoke the main thread encountered significant backlog or flow suppression in those threads. This release modifies the transmit logic to be invokeable by any thread (but only one thread at a time) to avoid this situation. The message receiving and invocation of the actor's `receiveMessage()` method still occurs only on the primary thread. No change in Actor implementation is indicated, and this should generally improve the functionality of any multi-threaded Actors.

Note that the use of Thespian Actors suggests that Actors rather than threads should be used for parallelism; this support is designed for Actors which import libraries where those libraries internally create multiple threads (e.g. kazoo).

- Modified recent async transport base changes to ensure that the maximum number of message sends are performed at any point in time. The previous release introduced a degradation wherein a single send could block the processing of other, independent message sends for a period of time; messages were still delivered but often not in a timely manner.
- Enhanced the thespianShell utility's simple source authority to propagate available `sourceInfo` field for loadable sources.
- Fixed a missing import error when resolving race conditions where multiple `globalName` actor requests resulted in multiple actors being created.
- Minimized traceback line-number offsets in loaded sources from an offset of 3 to an offset of 1 line number.
- Documentation updates, including a warning about the inability to exchange objects between different loaded sources unless conventionally serialized (see Source Loading Issues).

3.2.1 <2016-09-08 Thu>

- Added backward compatibility support for interacting with pre-3.2.0 Thespian instances without the `sourceInfo` argument.

3.2.0 <2016-09-08 Thu>

- Added `sourceInfo` argument to `ValidateSource`; allows informational tag to be associated with a loadable source for logging and status information. Thespian will automatically attempt to identify the filename for the `ActorSystem.loadSource()` operation and provide that on status requests.
- Updated handling for multithreaded Actors to allow any thread to `self.send()` messages (or generate logging output); receiving messages still occurs only on the main thread.

3.1

3.1.1 <2016-09-08 Thu>

- Better TCPTransport handling of connection attempts.

- Added missing import for exception case in async transport base.
- Explicit disable of ActorAddress hashing.
- Updated message for ActorSystem.tell() bad target address exception failure.

3.1.0 <2016-08-29 Mon>

- API and Public Changes
 - Adds the ability to allow specification of a "Process Startup Method" capability when starting Actor Systems to select the process creation technique for new Actors: "spawn", "fork", or "forkserver".
 - Updates the thespianShell.py utility to add Python3 compatibility and a simple Source Authority for accepting test actor loadable sources.
- Behavioral Fixes
 - Actors which create multiple internal threads are now supported. Thespian will still create Actors as separate processes if using a "multiproc..." system base, but if those Actors internal create threads (directly or through the use of a library), those threads will no longer cause failures or deadlocks when calling self.send() or logging.
 - Adds a small timeout period for TCP transport outbound connection initiation to accomodate busy or noisy networks.
 - Fixes a system shutdown completion failure management bug that formerly caused some Actors to never receive shutdown requests.
 - Adds timeouts to MultiprocessQueueBase transmits (matching functionality of the multiprocTCPBase and multiprocUDPBase) to avoid indefinite hangs on errors or blocks.
- Internal Bugfixes
 - Ensures that source unloaded notifications are not reciprocally sent between actor systems that would result in indefinitely repeated notifications.
 - Sets an upper limit on individual socket receive calls in the TCP transport to prevent starvation. This is only a limit in individual receives; the overall packet size can be larger than the receive and it will be reconstructed on the receiving side.
 - Retries more frequently in the TCP transport if waiting on an outbound transmit slot.
 - If the TCP transport encounters a connection reset while sending an ACK, this event is simply logged instead of throwing an exception.
 - Various test updates for parameterizing the various allowed time durations during the test.
 - Miscellaneous internal formatting and code cleanup.

3.0

3.0.4 <2016-08-12 Fri>

- Fix for initialization hardening in previous release.
- Optimize `__import__` interceptor to avoid repeated lookup failures.
- Extended actor creation startup allowance period to accomodate remote fetch of large sources.

3.0.3 <2016-08-11 Thu>

- Hardened initialization for environments where the fqdn, hostname, or dns lookup could cause `gaierror` exceptions.
- Fixed error reporting for actor creation failure exception.
- Internal code cleanup/enhancements and minor test updates.

3.0.2 <2016-07-29 Fri>

- More updates to tests for timing issues and adding idempotency of assert statements because `pytest` may re-execute them.
- An Actor call to `createActor` may specify a `sourceHash` that it does not have the source for; fixed to properly handle this case to defer the `createActor` to the Admin and remove the erroneous error log output.
- Only hook `__import__` for Python3.5 source loading; do not replicate all builtins.

3.0.1 <2016-07-26 Tue>

- Internal updates to tests for selection of available network ports for each test's admin address.
- Added test delay function for convenient centralization and standardization of testing delays and the ability to override, mock, or adjust the delay as needed.
- Updated `mark.skip` specifications for broader `pytest` compatibility.
- On loaded source unload, skip `None` (in-progress) entries in `sys.modules`.
- Optimize source load import intercept; only one definition per `sourceload` module is needed instead of per file.
- Updated thespian shell to handle non-hashable Actor Addresses.
- More information reported in Pending Actor timeout exceptions.

3.0.0 <2016-07-04 Mon>

- Actor Addresses are not hashable. Previously they were incorrectly marked as hashable, but they may be updated internally over time, and this internal mutability makes them non-hashable. This does not affect the ability to compare addresses for equality and inequality.
- Converted to using pytest for test framework; more information is available in the `thespian/test/00README.txt` file. These changes include significant test updates and enhancements, as well as broader coverage of all current system bases and configurations (e.g. Admin Routing and TX Only base settings).
- Documentation updates, including distinguishing this Thespian against unrelated efforts with overlapping similarities.
- Updated sourceloading support to handle uses of `__import__` directly (e.g. for sqlalchemy support).
- Better handling of socket accept failures in the TCP transport.
- Fixed ChildActorExited messages for unresolved local addresses to no longer be fatal issue.
- Updated Convention pre-registration to properly identify as a pre-registration and to send a Convention De-Register for pre-registered systems on system shutdown.
- Sending an ACK or NACK in the TCP transport is no longer capable of causing a hang in the transport on blocking sends or errors.
- Updated the common Capabilities definitions for the simpleSystemBase to match the other system bases.
- More regular throwing of exceptions for `ActorSystem().createActor()` requests for top level actors that cannot be satisfied.
- Asynchronous transports will immediately fail transmit requests if the outbound queue has reached an upper-limit threshold in size.
- If in transmit-only mode, fail expired requests appropriately.
- Fix wakeup-supporting transports to always check for expired wakeups and immediately-processable events even if there is no positive non-zero time duration for transport processing.
- Revert signal handling introduced in version 2.5.8: internal Python issues (e.g. <https://bugs.python.org/issues14976>) made the initial approach buggy. Newly engineered signals support will be added in a forthcoming release.
- Updated Convention Notification handler removal for common code to ensure the same treatment is given for all cases of handler removal.

2.x**2.6****2.6.1 <2016-05-20 Fri>**

- Fix simpleSystemBase wrapping of DeadEnvelope for dead letter re-routing.

- Fix SIGUSR1 status output for Windows, which has no SIGUSR1 (and therefore does not support this functionality).

2.6.0 <2016-05-18 Wed>

- Functional Changes
 - Added status output to thesplog on SIGUSR1 to an Actor. This is useful for scenarios where the Actor is not responding to regular queries (e.g. network queue blockages, etc.).
- Behavioral Fixes
 - The simpleSystemBase now properly supports system capability updates and validates the actorSystem-CapabilityCheck results when creating Actors.
 - The httpserver example is fixed for handling requests with no QUERY_STRING portion.
 - Do not try to restart the logger if the system is being shutdown.
- Internal Bugfixes
 - Many tests update to use independent admin ports to avoid inter-test influences.

2.5

2.5.10 <2016-05-03 Tue>

- API and Public Changes
 - Added a Websocket Actor example and the Actorize decorator contributed code from Paul Jimenez.
 - Added ThespianWatch and WatchMessage functionality.
- Behavioral Fixes
 - Update logging if convention leadership is changed for logging via new leader.
 - Restart the logger if it exits.
 - Update signal handling to better interrupt wait cycles in transports, resulting in handling signals more immediately.
 - Ensure actor checks/restarts caused by updating Actor System capabilities only affect actors on the updated system.
 - Fixed bug on address forwarding that caused transmit lockups when the forwarded message exceeded the pending transmit limit.
- Internal Bugfixes
 - Update the logging to use a global transport instance to ensure that children inheriting threads will use the proper transport instance.

- Add small wait to allow child exit information to propagate and be handled.
- Update the admin for handling child exit signalling to provide better detection of children that are no longer running.
- Miscellaneous test updates to ensure distinct Actor Systems used for different tests to avoid crosstalk failures.
- Better handling of closed/bad file descriptors in the multiprocTCPBase transport to handle remote closures, etc.
- Use a thread-safe Queue in the async transports to ensure that transmits queued by alternate threads (e.g. log messages) only enter the core transport work loop via the main thread.
- TCP transport cleanups from Paul Jimenez.
- Fixed wakeup transport handling for TransmitOnly mode.

2.5.9 <2016-03-21 Mon>

- Fix signal handling introduced in 2.5.8 release to support Windows and other operating systems that don't define the full set of standard Unix signals.

2.5.8 <2016-03-21 Mon>

- Added protection to prevent simpleSystemBase internal addresses from being killed (Issue #5).
- Updated documentation to clarify the status, availability, and intended usage of ActorSystemMessage messages (Issue #6).
- Updated documentation to describe the potential for Message Mutability for some ActorSystem bases as a tradeoff for performance (Issue #7).
- Added internal atexit and signal handling functionality to attempt normal Actor shutdown (via ActorExitRequest message delivery) in all circumstances.
- Added internal SIGCHLD signal handling to send ChildActorExited messages to parent Actors in cases where the child process was unable to send that message before exiting.
- Updated documentation to clarify which ActorSystem Capabilities are read-only (for each system base).
- Modified internal loaded source hash tagging for better handling of circular import references in loaded sources.

2.5.7 <2016-03-14 Mon>

- Updated loaded source lookup failure to throw ImportError exception instead of BadZipFile exception for compatibility with importing code handling the former.

2.5.6 <2016-02-11 Thu>

- Fixes for detection of socket closure in the TCPTransport: corrected errno reference and add handling of additional errors (for Windows and Mac OS/X).
- Update TCPTransport to ensure select() is never called with empty select lists (needed for Windows).
- Added small delay in ActorSystem shutdown to allow shutdown-related communications to complete.
- Update source loading tests to allow Source Authority registration before commencing test.
- Selection of "temp" directory for writing logs adjusted to be OS-aware.
- Added logging for actor instantiation failures (normal logging in addition to previously-existing internal logging).
- Updated default logging for thespian tests to support Windows which cannot inherit an open logging file descriptor in forked children.

2.5.5 <2016-02-05 Fri>

- Added fix for closing open TCPTransport sockets to a remote Admin that is known to be unavailable (i.e. on explicit or time-based de-registration).
- Ensure that all sends are routed through the local admin if the local environment is TX-Only ("Outbound Only": True capability) for TCPTransport-based communications.

2.5.4 <2016-02-01 Mon>

- Calling loadSource() multiple times for the same source does nothing instead of unloading and reloading the source. This provides more consistent behavior, especially when working with objects instantiated from the original source.
- Fixed possible scenario where logging of errors from TCP transport would encounter further problems.
- Test updates.

2.5.3 <2016-01-28 Thu>

- Fixed hash representation of TCP routed and tx-only addresses to provide consistent hash values and match the equality-sense of the addresses.
- Better convention management, especially for pre-registered members.
- Minor update to ActorAddress string representation for TCP routed addresses.

2.5.2 <2016-01-13 Wed>

- Enhanced and fixed logging messages, including severity indications.
- Formatting updates in the design document.
- Update allowable multi-system actor response times in tests.
- Limit status key/value output column alignment width to 40 characters.
- Added TCPTransport idle sockets to status reports.
- Fix multiple TCPTransport transmit retry handling and re-queueing bugs.
- Enhance TCPTransport to gracefully handle receive of a remote message while waiting for an ACK of a locally transmitted message.
- Update TCPTransport for full address/target re-evaluation on message forwarding to properly handle dead targets and related scenarios.
- Immediate cancel of all pending transmits to a remote target on Connection Refused (remote target is dead).
- Fixed TCPTransport to never use a negative delay (occurs if pending wait time on an event has already expired) and use a zero delay instead.
- No socket-level timeout on connects from TCPTransport; timeout management is done internally to TCPTransport.
- Allow connection to Admin even if the reported address is different than the expected address. This allows support for systems that change addresses or which have multiple viable addresses.

2.5.1 <2015-12-16 Wed>

- Fixed sourceload path handling: zipfiles use posix paths only, but running loaded sources on Windows with previous versions attempted to use Windows path specifications.
- Increased level of logging from sourceload failures to ERROR from INFO.

2.5.0 <2015-12-16 Wed>

- Allow value specification with the `@requireCapability` decorator.
- Add the `preRegisterRemoteSystem` and `deRegisterRemoteSystem` Actor API methods. These assist in identifying remote Actor Systems that might not normally have connectivity. These are most useful with the 'Outbound Only' capability (see below).
- Updated the TCP system base to persist and re-use open socket connections; previous versions would always close the socket after sending or receiving a single message. This change can increase network efficiency and reduce latency. This change is backward compatible with previous Thespian 2.x versions and will devolve into the older disconnecting-mode of operation.

- Added "Admin Routing": True capability specification (TCP system base only). The presence of this capability indicates that all traffic routed to any Actor on that Actor System must be forwarded via the Admin on that Actor System (Thespian handles the forwarding automatically when this capability is set). This configuration is useful when two Actor Systems are separated by firewalls that only allow a limited set of ports to communicate through the firewall; setting the Admin port to one of the firewall-allowed ports and setting the "Admin Routing" capability will route all traffic via the Admin (at a small performance penalty). Enabling this setting is *not* backward compatible with previous Thespian 2.x versions. This setting should be considered *Beta* functionality at this time.
- Added "Outbound Only": True capability specification (TCP system base only). The presence of this capability indicates that the current system cannot accept incoming connections. This setting implies "Admin Routing": True for this same system. This setting causes all Actor traffic to be routed through the Admin, and the Admin will open and leave open persistent connections to all remote Admins. This setting is useful where a firewall or VPN configuration will only allow outbound connections, but it incurs additional performance penalties and potential delays while an outbound connection is being re-established. Enabling this setting is *not* backward compatible with previous Thespian 2.x versions. This setting should be considered *Beta* functionality at this time.
- The "Thespian ActorSystem Version" for the TCP system base has been updated to "2" from "1" to indicate the above changes.
- All systems that have attempted to create an Actor will be remembered to avoid infinite creation loops.
- Logging updates and enhancements.
- Enhance shutdown by ensuring the exiting status is set before shutting down children to handle their responses appropriately and avoid race conditions.
- Show statistics values before labels in the status output for better readability.
- Design document updates.
- The simple system base now properly implements the `wakeupAfter()` delays. It will not hang if only wakeups are pending, but it will deliver them after at least the appropriate delay and relative to `tell()`, `ask()` and `listen()` timeouts.
- All `loadActorSource()` calls are ignored if there is no registered Source Authority. This changes the disposition of an Actor System to `secure/protected` by default.

2.4

2.4.4 <2015-12-16 Wed>

- Removed TCP connect timeout that overrode non-blocking mode.
- Fixed TCP address equality comparison when zero port specification compared. A port of zero matches any other port if the address portion is the same; previously this only worked for comparing to known local addresses, not arbitrary addresses when the port was 0.
- Fixed error text when using `tell()` with a bad Actor Address (previously identified the wrong call).
- Ensure Actor System `createActor()` calls are not aborted prematurely by leftover responses to previous operations (e.g. timeout-unclaimed `ask()` responses).

2.4.3 <2015-11-18 Wed>

- Ensured that PoisonMessage sending doesn't recurse indefinitely.
- Added a missing newline in status output formatting.
- Added a descriptive failure output messaging for IP-based address connection failures. This helps with common mistakes, such as specifying the "Admin Port" as a string instead of an integer.
- Fixed a bug where calling `ActorSystem().shutdown()` multiple times would throw an exception.
- Fixed a bug on multi-level import references (e.g. `import x.y.z`) in sources loaded by `loadActorSource()` operations.
- Fixed a bug where old actors checking in with a restarted ActorSystem caused exceptions.
- Fixed a TCP transport bug that caused transmits to be abandoned instead of being retried after the back-off timer expired.
- Updated default thespian.log logfile creation to honor `$TMPDIR` if it is set and only default to directing logging to `/tmp` if it is not set.
- Added internal `wait()` call for logger child process on shutdown to avoid defunct loggers.
- Removed a debug `printf` in the `simpleSystemBase`.

2.4.2 <2015-10-26 Mon>

- Delay thesplog location determination until usage to allow startup code to adjust location.
- Better daemonization of multiproc Admin on Unix.
- Ensure `filenums to close` is passed as a list not a Python3 iterator when creating multiproc children (bugfix).

2.4.1 <2015-10-20 Tue>

- Updated to allow packets not blocked on waiting for an address resolution to be sent even if previous packets are still waiting for that resolution.
- Fixed documentation uploads.

2.4.0 <2015-10-18 Sun>

- Add `loadActorSource()` and `unloadActorSource()` to Actor API (duplicating existing Actor System API functionality).
- Internal: use distributed `importlib` whenever possible.
- Fix stats response for Actor Systems for Python3.

- Allow optional error strings in reporting actor creation failures.
- Better capture and reporting of loaded source instantiation errors.
- Miscellaneous test and logging updates.

2.3

2.3.0 <2015-10-04 Sun>

- Added ActorTypeDispatcher enhanced base class to facilitate dispatching of received messages based on message type.
- Ensure exceptions occurring while writing to Thespian file log do not cause failures.
- Protect against exceptions occurring when comparing messages for equality.
- Updates to testActorSystemRestart.
- Update "Thespian Generation" common Capabilities to version (2,3). Previous 2.2.0 release erroneously did not update this value.

2.2

2.2.0 <2015-09-27 Sun>

- Fix internal links in Using Thespian document.
- Add support for multiple convention notification registrations.
- Allow ActorSystem logDefs to be specified as False to suppress any logging configuration; especially useful for test log capture modes.
- Better ActorSystem startup processing to distinguish between a re-specification of the current base and a new base, and added the transientUnique startup argument to get a plain instance (a non-global-singleton instance).
- Added missing description of the ActorSystem shutdown() operation in the Using Thespian document.
- Added ability to set the maximum thesplog logging file via the THESPLOG_FILE_MAXSIZE environment variable and changed the default to 50KB.
- Added Using Thespian documentation description of logging with the THESPLOG_FILE and THESPLOG_FILE_MAXSIZE environment variable controls.

2.1

2.1.6 <2015-09-27 Sun>

- Fix imports in tests and actors.py for multiple Python versions.
- Add missing timeout to test to prevent hang.

2.1.5 <2015-09-20 Sun>

- Adds support for Python 3.3 and Python 3.4 versions, including the use of the PyPy interpreter.
- Adds the `listen()` API call (`ask()` is `tell() + listen()`)
- Log Actor exception through normal logging channels as well as via Thespian internal logging output.
- The Thespian shell loads the TestActor via loadable sources.
- The `simpleSystemBase` will always apply default logging.
- The `loadActorSource()` call can now take a file-like object as well as a filename.
- Added the Thespian Developer's document.
- Added the Thespian In-Depth Introduction document.
- Added the Thespian Index (main pages) document.
- Added the Thespian Releases document.
- Internal updates to more gracefully handle shutdown scenarios and avoid impossible retransmits.
- Testing, documentation, and setup file updates.

2.1.4 Initial Public Release <2015-08-31 Mon>

The 2.1.4 Release represents the first publicly available release of Thespian. It includes the following features:

- System Bases: `SimpleSystemBase`, `MultiprocTCPBase`, `MultiprocUDPBase`, `MultiprocQueueBase`
- Support for Loadable Sources

2.0

The 2.0 Release was developed and used internally by GoDaddy.

1.x

The 1.x Release series was developed and used internally by GoDaddy

