



Release Notes

Thespian Python Actor System

By: Kevin Quick <kquick@godaddy.com>

2016 Dec 08 (#3.5.1)

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.

3.x

3.5

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. <http://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 Mon>

- 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 Mon>

- 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