The PMIx Standard Governance Rules

Version 1.6 (Draft)

Created on February 15, 2022

This document describes the PMIx Standard Governance Rules, version 1.6 (Draft).

Comments: Please provide comments on the PMIx Standard by filing issues on the document repository https://github.com/pmix/governance/issues or by sending them to the PMIx Community mailing list at https://groups.google.com/forum/#!forum/pmix. Comments should include the version of this document that you are commenting about, and the page, section, and line numbers that you are referencing. Please note that messages sent to the mailing list from an unsubscribed e-mail address will be ignored.

Contents

1 PMIx Standard

PMIx is an application programming interface (API) standard to provide High Performance Computing (HPC) libraries and programming models with portable and well defined access to commonly available distributed computing system services.

1.1 Governance

The PMIx Standard is governed by an Administrative Steering Committee (ASC) comprised of representatives from a wide range of research, academic, and industrial organizations. Membership in the ASC is determined by participation in the PMIx Standard meetings and a vote of the ASC. Any organization meeting the membership criteria can become an ASC *Member*. An organization may refer to any entity, e.g., a corporation, university, national laboratory, or an individual person that is not employed by another Member organization (as detailed below). Each Member shall be represented by a designated individual (the *Representative* who is authorized to vote on behalf of the Member. If desired, a Member may designate an *Alternate* who is authorized to vote on behalf of the Member on occasions when the Representative is not present.

1.1.1 ASC Duties and Organization

The ASC performs the following functions:

- Define the mission of the PMIx Standard.
- Plan release timelines for versions of the PMIx Standard.
- Vote on all issues requiring community consensus (e.g., acceptance of proposed modifications to the Standard), one vote per Member as cast by their Representative or Alternate. In addition to voting on proposed modifications, the ASC can be requested to vote on any issue submitted by a Member for consideration.
- Assemble working groups to address specific areas of interest to the community.
- Nominate and approve the appointment of individuals to fill the roles of Release Manager (per release) and Secretary, as described below.

The ASC is led by two *Co-Chairs* responsible for organizing and conducting the quarterly meetings. This includes:

- Announcing and finalizing meeting agendas.
- Chairing the quarterly meetings
- Ensuring timely publication of the minutes of the meeting, and tallying votes.

Co-Chairs are elected from amongst the ASC members during the last quarterly meeting of each calendar year (using the voting system described below). The newly elected Chairperson shall take office at the beginning of the following calendar year, thus allowing overlap with the outgoing Chairperson for efficient transition of active work items. Co-Chairs serve for a period of two years, with seats filled on alternate years - i.e., one seat will be elected during the last quarterly meeting of each even year, and the other seat will be elected during the last quarterly meeting of each odd year. There is no restriction on the number of consecutive terms a given individual may serve - however, it is expected and desirable that the positions rotate amongst the Members. Likewise, there is no requirement that a Co-Chair be the Representative or Alternate for their Member. Co-Chairs serving as Representative or Alternate for a Member may vote on behalf of the Member subject to the one vote per Member limit.

In the event a Co-Chair decides to step down from the position before the completion of the two-year service period, the ASC will hold an election for a replacement Co-Chair at the next quarterly meeting. The newly elected replacement Co-Chair will serve for the remainder of the time of the two-year service period of the original Co-Chair. The replacement Co-Chair may seek election in subsequent terms.

1.1.2 ASC Participation

Participation in the consortium is both formal and informal:

- Informal participation is encouraged by working with the PMIx Standard community and/or one of the PMIx implementation libraries in an *ad hoc* manner, such as (but not limited to) engaging in discussions on the mailing lists, reporting testing results, providing feature requests and bug reports, filing pull requests for proposed modifications, etc. *Participants* do not have voting privileges on administrative matters, but are encouraged to attend the quarterly meetings and participate in those discussions. Organizations must attend meetings as Participants in order to qualify for elevation to full Member status, as described below.

- Participants become eligible for elevation to Member status upon having at least one representative attend two quarterly meetings within the prior twelve months. All new Members must either request membership or be nominated by an existing ASC Member, and be approved by the ASC according to the voting rules described below. Membership confers representation on the ASC, thus granting the right to participate in formal votes, along with public acknowledgment for the entity as being a Member of the PMIx Standard ASC on the PMIx web site. Each Member organization shall appoint one official Representative with authority to vote on their behalf - if desired, an Alternate may also be designated. The Representative for an organization is responsible for notifying the ASC in advance of a temporary alternate should the Representative (and the Alternate, if designated) not be attending a meeting in order to confer voting rights to that individual.

  - Note: Individuals may also become Members so long as their direct employer is not a current Member of the ASC. For these purposes, a consultant is not considered a direct employee - thus, someone consulting for a current ASC Member is eligible to become an individual Member should they choose to do so. Individual Members who subsequently join an organization that is a Member will be absorbed into the larger organization's membership and will surrender their individual voting rights. Individuals requesting membership but are affiliated with multiple ASC Members can be granted individual Member status by vote of the ASC. This is intended to preclude undue influence in voting procedures by an organization with multiple individual Participants.

- A Member is removed from the ASC (a) upon having failed to have at least one representative attend two quarterly meetings in the prior twelve months, or (b) by vote of the other ASC members per the voting rules defined below. Attendance by a non-voting member of the organization (i.e., someone not officially designated as an alternate for voting purposes) counts towards meeting this requirement.

1.1.3 ASC Meetings

The ASC meets quarterly, with at least one quarterly meeting each year to be held "in-person" at an agreed-upon location unless the ASC Members vote otherwise. A vote to "virtualize" the in-person meeting can only apply to that given year. Other quarterly meetings may be held by teleconference in accordance with the desires of ASC Members. Participants are welcome to attend any meeting in a non-voting capacity. ASC quarterly meetings are conducted once per calendar quarter with a preference for a time during the first month of the quarter. Precise meeting dates are determined by the ASC Members in accordance with their schedules, subject to the voting rules described below. Subsequent changes to a given meeting's dates must be determined no later than two quarterly meetings before the one being changed.

The Co-Chairs shall include links to each proposed modification to the Standard up for consideration during that meeting in the published agenda.

ASC quarterly meeting agenda items for reading and vote must be submitted to the Co-Chairs and announced on the PMIx mailing list at least four weeks in advance of the meeting. Approximately four weeks in advance of the meeting the Co-Chairs will announce the tentative agenda for review. The community has until two weeks before the quarterly meeting to review and comment on the agenda and PRs scheduled for vote or reading. During the review period, the contents of the agenda and PRs scheduled for vote or reading may be changed. The agenda will be finalized two weeks before the quarterly meeting at which point no further additions or modifications will be accepted. A PR posted for vote or reading in the ASC quarterly meeting must not have any modification of the content of the branch after the agenda finalization deadline. Comments on the PR are welcome at any time, but the content of the PR must be frozen to allow for community review of the stable text. For a formal reading or vote, it is required that a PDF rendering of the changed document be sent to the PMIx mailing list and to the Co-chairs for distribution in the final agenda.

1.1.4 Modification of Governance Rules

Proposals to modify the PMIx governance document begin with the filing of a GitHub Pull Request (PR) in a Draft PR state and marked with a "Governance" label until ready for submission for review. This PR shall serve as the vehicle for capturing all discussion over the proposal. When ready for review, the author(s) shall add the "Proposed" tag to the PR, thereby requesting that the matter be placed on the agenda of the next quarterly meeting (subject to the agenda finalization deadline). The *Straw Poll* mechanism described below is utilized to provide an initial screening of the proposal. Passing of the Straw Poll *does not* confer approval of the proposal, but serves instead as a means for quickly pushing back proposals that lack significant support. This is intended to both protect the time of ASC Members and author(s), and to help guide author(s) wishing to continue pursuit of the proposal.

Upon successfully completing the Straw Poll, the Co-Chairs shall label the PR as "Eligible" to indicate that the proposal is eligible for formal consideration by the ASC. Given the significance attached to governance, formal approval of changes requires that the ASC conduct a full presentation on the proposal at two consecutive quarterly meetings. The timing of the second meeting may be delayed at the request of the author(s) or ASC Members, subject to approval by the ASC. At the conclusion of the first meeting, a formal vote of the ASC shall be held (per the voting rules below) to determine if the proposal merits further consideration. The Co-Chairs shall set a GitHub label indicating the result as either "First Vote Passed" or "Pushed Back".

At the conclusion of the second meeting where the proposal is up for consideration, another formal vote of the ASC shall be held (per the voting rules below) to determine final disposition of the proposal. The Co-Chairs shall subsequently set a GitHub label indicating the result as either "Accepted" or "Pushed Back". Accepted proposals can be committed by the Co-Chairs in accordance with their schedule, but take effect immediately (i.e., during the same meeting in which they receive final approval).

Proposals that are pushed back at any point can be modified and resubmitted by the author(s). However, the updated PR must pass through the complete approval process from the beginning to ensure full consideration of the revised request. A new PR for this purpose can be submitted if the author(s) so desire, but must be linked to the original PR in order to retain full information regarding prior concerns and suggestions.

1.1.5 General Voting Rules

The philosophy and goal of the ASC is to reach consensus on all, if not most, issues. In the event that full consensus cannot be reached, the ASC will use the following rules:

- Each active ASC Member will receive one vote. All votes are required to pass with 2/3 majority. A quorum (>50%) of Members must be present (either physically or virtually). Abstentions will not be counted in the total votes when calculating whether there is a 2/3 majority.
- A Member may submit a vote *in absentia* to the ASC Co-Chairs in advance of the meeting if attendance by their Representative or designated Alternate is not possible (e.g., a quarterly meeting that falls during a national holiday period), but each ASC Member can only cast one vote. This prevents a Member from voting on behalf of its own entity and acting as a proxy for another Member. *In absentia* votes are counted by the Co-Chairs as if the Member were present (thus counting towards the meeting's quorum requirement), but are recorded as having been voted *in absentia* in the official tally.

Revision Exception votes are special votes for PRs that have been modified after the review period deadline before a quarterly meeting before a first vote or after having passed a first vote. Revision Exception votes may be called by the proposer to evaluate whether the ASC is willing to proceed voting on the PR including the modifications. The idea behind Revision Exception votes is to maintain forward progress on PRs even if changes are made after the quarterly meeting deadlines or after a first vote. Normally, changes acceptable for Revision Exception votes are relatively minor and do not change the fundamentals of the PR. However, if a quorum of ASC voting-eligible members all vote affirmatively for a Revision Exception vote, then the regular vote for the PR may proceed for the text of the PR including the modifications. A quorum of ASC voting-eligible members must be present and vote in the affirmative without any negative votes for the Revision Exception vote to pass. If the Revision Exception vote does not pass, the proposer must choose to re-read the PR with the modifications or to continue the normal vote without the modifications.

Straw Polls are used in several places as a means of gauging community support for a proposal - e.g., in assessing acceptance for a new Provisional API. Straw Polls are conducted on an informal basis through the placement of emojis on a *Straw Poll Comment* of an Issue or PR. The poll comment consists of the following text:

```
Please use emoji reactions ON THIS COMMENT to indicate your position on this proposal.

    * You do not need to vote on every proposal
    * If you have no opinion, don't vote - that is also useful data
    * If you've already commented on this issue, please still vote so
          we know your current thoughts
    * Not all proposals solve exactly the same problem, so we may end
          up accepting proposals that appear to have some overlap

This is not a binding majority-rule vote, but it will be a very
significant input into the corresponding ASC decision.

Here are the meanings for the emojis:

    * Hooray or Rocket: I support this so strongly that I
          want to be an advocate for it
    * Heart: I think this is an ideal solution
    * Thumbs up: I'd be happy with this solution
    * Confused: I'd rather we not do this, but I can tolerate it
    * Thumbs down: I'd be actively unhappy, and may even consider
          other technologies instead

If you want to explain in more detail, feel free to add another
comment, but please also vote on this comment.
```

1.1.6 Release Managers

One or more Release Managers are assigned per release series of the PMIx Standard. Coordination and scheduling of releases in the series is delegated to the Release Manager(s) by the ASC, subject to guidance regarding needs on the part of Members. Release Managers have the following duties:

- For each release, lead the establishment of release criteria by the ASC. This can include a list of proposed modifications to be decided upon prior to release and target release dates
- Shepherd progress towards completing the schedule through interactions with proposal authors
- Maintain the Revisions section of the document to ensure accurate communication of what has changed in each release

Release Managers shall be selected from within the community of Members, but are not required to be a Representative or Alternate. Note that Release Managers are not precluded from proposing modifications to the Standard (including to the series under their management) during the course of their assignment, nor are they prevented from voting if they also serve as Representative or Alternate for their Member. In addition, Release Managers are not responsible for promoting acceptance of any proposal nor contributing to the completion of a proposal in order to meet schedule, and may resolve any merge conflicts resulting from commit of approved changes without further process.

The Release Managers shall notify the Co-Chairs when a minor release candidate is complete and ready for publication. The Co-Chairs will add an item to the agenda of the next quarterly meeting (subject to the agenda finalization deadline) for final approval to publish the document. Once the ASC has voted its approval, the Release Managers shall post the new release on the community's web site and send an email announcing the release to the community mailing list. Minor releases, therefore, can occur on an at most a quarterly basis.

Major release candidates must be approved during the ASC's annual face-to-face meeting. Release Managers must notify the Co-Chairs of a major release candidate a minimum of two quarters before the target approval meeting. The Co-Chairs will add an item to the agenda of each quarterly meeting for discussion and review of the document, and on the face-to-face meeting for vote on final approval. Once the ASC has voted its approval, the Release Managers shall post the new release on the community's web site and send an email announcing the release to the community mailing list. Should the ASC not approve the release, subsequent release candidates can be considered at upcoming regular ASC meetings - i.e., the major release candidate does not have to wait for the next annual face-to-face meeting for reconsideration. However, no subsequent major release may begin the approval process until after the current candidate has been published. Major releases, therefore, can occur no more than once per year.

1.1.7 ASC Secretaries

The ASC has two Secretary positions that are responsible for taking and publishing minutes of quarterly meetings, and maintaining the PMIx Standard Web site. Other duties may be defined by the ASC as required. Secretaries are elected from amongst the ASC members during the last quarterly meeting of each calendar year (using the voting system described above). The newly elected Secretary shall take office at the beginning of the following calendar year, thus allowing overlap with the outgoing Secretary for efficient transition of active work items. ASC Secretaries serve for a period of two years, with seats filled on alternate years - i.e., one seat will be elected during the last quarterly meeting of each even year, and the other seat will be elected during the last quarterly meeting of each odd year. There is no restriction on the number of consecutive terms a given individual may serve - however, it is expected and desirable that the position rotate amongst the Members. Likewise, there is no requirement that the Secretary be the Representative or Alternate for their Member. A Secretary serving as Representative or Alternate for a Member may vote on behalf of the Member subject to the one vote per Member limit.

1.2 ASC Guidance

This section provides general guidance for the PMIx ASC and participants. The rationale behind providing this guidance is to outline philosophy and best practices the ASC should follow in maintaining the PMIx Standard.

1.2.1 Release Schedule Guidance

As described in Section 1.1.6, the schedule of major releases of the PMIx Standard is determined by the Release Managers in coordination with other members of the ASC. There is no prescribed major release schedule that the Release Managers must follow. However, the Release Managers will prioritize timely releases of substantive changes to the Stable entries in the PMIx Standard. Substantive changes include: deprecated or removed APIs, keys, or other text in the standard; alterations of APIs or keys; and alterations of text that result in changed meaning or clarifications. The Release Managers will prioritize major releases of the standard over waiting for additional changes that are in progress but not yet accepted by the ASC.

1.2.2 Minor Document Changes Guidance

The intention of the rigorous process outlined in Section 1.3 for proposed changes to the PMIx Standard is to ensure quality and consistency of the interface and the overall document. However, in some cases, PMIx participants may propose minor changes to the document that should not require the full process required for substantive changes. Examples of these minor changes include fixes for typographical errors (“typos”), missing or erroneous punctuation, and formatting changes.

In the case minor changes are proposed, the proposer should create an issue and pull request as described in Sections 1.3.2 and 1.3.3 and then bring the changes to the attention of the ASC Co-Chairs and Release Managers. The Co-Chairs and Release Managers will evaluate the proposed changes to determine whether they are minor or substantive. Additionally, all PMIx participants are encouranged to examine and comment on the changes in the pull request to aid the Co-Chairs and Release Managers in their decision. If all Co-Chairs and Release Managers determine that the changes are minor and do not alter the meaning of the current standard text, then the changes do not require the full process and the changes can be applied to the document in the next release. Alternatively, if any of the Co-Chairs or Release Managers determine that the changes do or could alter the meaning of the current standard text, the changes will be handled using the full rigorous process as described in Section 1.3.

1.3 The Standardization Process

Contributions of any form (e.g., use cases, questions, suggestions) are encouraged from anyone interested in the evolution of the PMIx Standard. The community relies on Issues and Pull Requests to discuss changes to the standard for a few reasons:

- It provides an archive of the conversation around a change to the standard that can be vital in understanding the rationale for the change if that conversation needs to be revisited in the future.

- It facilitates asynchronous conversation from a geographically distributed user-base. To that end, any discussion on the teleconference about specific Issues or Pull Requests will be noted as a comment to that Issue or Pull Request.
- The GitHub mechanism serves as a consensus building medium for accepting changes once all dissenting opinions and questions have been resolved, providing an archive of the responses to those concerns and those organizations participating in the process.

Templates are provided to assist in writing an Issue and/or Pull Request that includes an appropriate level of information for that type of request. Questions may be answered by anyone and participation from the broad PMIx community is encouraged and welcomed, as are suggestions for clarifications and corrections to the Standard. Feature requests are best presented in the form of a use-case as this assists both the community's understanding of the justification for providing the feature and in assessing the feasibility/magnitude of effort required to provide it.

Proposed modifications to the Standard can come in several forms:

- Addition of one or more new APIs and/or key-value attribute definitions
- Changing the Standard classification of a current API and/or attribute
- Deprecating an existing API and/or attribute
- Clarifications and corrections to existing Standard language

The Standardization Process remains essentially the same regardless of the type of modification being proposed, although the speed at which a modification progresses through the process may vary considerably based on the magnitude and controversial nature of the proposed change. Discussions regarding a proposal are conducted either on the Issue/PR itself or during the regular community teleconference (and captured as notes on the Issue/PR). Teleconferences provide a high bandwidth discussion format and valuable synchronization mechanism to guide progress. Non-issue/PR related notes from the teleconferences are archived on the PMIx community's wiki.

Note that a proposal can be withdrawn by the author(s) at any time for any reason up to the time the proposal is committed to the Standard, assuming acceptance.

1.3.1 Classes of Standard

The PMIx Standard is composed of a combination of APIs and key-value attributes. APIs are intentionally designed to be "lightweight" - i.e., the API signature is generic with a minimal number of fixed parameters in its signature. With few exceptions, APIs are required to include an array of key-value attributes in their signature to facilitate behavioral extensions without modifying the API signature. Thus, PMIx attempts to minimize deprecation and modification of APIs as use models evolve over time.

PMIx APIs and attributes are grouped into three classes:

- *Provisional* entries in the Standard are officially part of the Standard but are not yet guaranteed to be stable - e.g., an API signature and/or attributes it supports may be changed, or the entry can be completely removed if subsequent evaluation proves negative. New provisional entries can be added to any minor release of the Standard. Subsequent changes to an entry requires warning in at least one minor release of the Standard document before acceptance - i.e., a proposed change to a provisional entry announced in version A.1 of the PMIx Standard may be committed in version A.2 of the Standard. This includes deprecation of an existing provisional entry. Thus, the intent is to provide a relatively easy mechanism by which proposals acceptable in principle to the community can enter the Standard prior to being fully stabilized, thereby allowing early adopters an opportunity to explore the proposed capabilities.

  All proposed API and attribute additions to the PMIx Standard enter the PMIx Standard under Provisional status. Attributes in the Provisional class may only be designated as "optional" by an API - i.e., an API definition cannot designate a Provisional attribute as "required". Users, however, are always free to "require" that an attribute be supported when calling the API, thus directing the API to respond with a "not supported" error should the attribute not be supported by it. Attributes that are intended to become designated as "required" by an API when elevated to Stable status shall be clearly designated as such in the proposal.

Due to the flexible nature of the Provisional class, no guarantees are made with respect to backward or cross-version compatibility for entries in this class.

- *Stable* entries are permanent, non-changing parts of the PMIx Standard subject solely to deprecation. In this class, full guarantees are provided with respect to backward and cross-version compatibility beginning with the version upon which the entry was granted Stable status. No guarantees are made with respect to versions that included the entry during its Provisional status.

  Stable APIs may continue to define and/or adopt new supported key-value attribute pairs, thereby extending the *behavior* of a stable API without modifying its signature. As previously stated, attributes in the Provisional class can only be specified as "optional" - therefore, *required* attributes used by APIs in the Stable class must also reside in that class. A Stable API that wishes to utilize a Provisional attribute must either first seek elevation of the attribute to the Stable class so it can be "required", or must designate that attribute as "optional". This is necessary to ensure that all Stable parts of the Standard are capable of meeting the compatibility requirements. Provisional APIs are, of course, free to utilize any Stable attribute in addition to anything in the Provisional class.

- *Deprecated* entries are slated for removal in an upcoming major release of the Standard. Deprecation warning must be provided for a minimum of one major release - i.e., an entry that is marked for deprecation in release A.0 cannot be removed from the Standard until at least the A+2.0 release. Extensions to the deprecation deadline can be granted on a case-by-case basis upon vote of the ASC.

1.3.2 Initial Discussion

The process of amending the PMIx Standard begins with the opening of a *GitHub Issue* describing the proposed change. The Issue template used and the degree of detail provided in the issue should be commensurate with the magnitude of the proposed modification - e.g., a spelling correction would use the "Questions & Clarification" template and require very little justification, while a new API or deprecation of an existing API might use the "Use-Case" template and reasonably be expected to be accompanied by a more detailed justification. Use of the template corresponding to the type of modification being proposed is helpful to expedient processing of the proposal - if in doubt as to the one to use, please ask.

The Issue provides a discussion forum for the contribution without necessarily specifying precise changes to the PMIx Standard document. Once an Issue is ready for discussion, it is helpful, though not required, to send a message to the PMIx Forum mailing list to raise awareness. Note that an Issue may not result in an actual change to the Standard - e.g., it might serve simply as a focal point for community discussion regarding a potential use-case or problem arising from a use-case.

If a change to the PMIx Standard is included in the issue (or results from the discussion in the issue), then a Pull Request must be created (see below). Note that multiple Pull Requests may reference a single Issue as authors consider alternative approaches to addressing the Issue. The Issue will be closed once the associated PRs have been resolved (either through by being accepted, withdrawn, or rejected) or once the participants are satisfied with the conversation.

1.3.3 Proposed Modification

When a specific change to the PMIx Standard has been identified a PR must be opened with a reference made to the corresponding Issue(s). When the PR is opened it should be placed in a Draft PR state until the following conditions are met:

- The contents of the PR are ready for review and broad discussion with the PMIx community. Note that discussion over the PR is encouraged at all times even while in draft form.

- An open-source implementation of the proposal is available and a reference to the implementation included in the PR. A complete implementation is not necessarily required to meet this condition - e.g., pseudocode detailing the execution path supporting the proposed modification can be provided. More complete prototype implementations may be requested by the PMIx community when proposals involving significant changes in behavior are made.

Only new commits are permitted on the branch once the Draft PR designation has been removed. The branch associated with the PR should not be squashed or force pushed after this point to preserve the timeline of changes in

relationship to the discussion that may occur on the PR. Note that PRs that are squashed or force pushed remain eligible for consideration, but may increase the difficulty of obtaining consensus from the community.

Discussion of the PR will continue until all dissenting opinions and questions have been addressed, upon which the PR may be put forward for a formal decision by the ASC. Note that addressing dissenting opinions does not necessarily mean achieving complete resolution on them. A dissent is not considered to have "veto" power over a proposal, but must be addressed to a degree that convinces the majority of ASC Members.

1.3.4 Provisional Acceptance

The process for acceptance to Provisional status begins when the author(s) assign an "RFC" label to the GitHub PR, add the Straw Poll Comment to it, and send an announcement of the PR to the PMIx community mailing list requesting consideration of the PR for Provisional status. The announcement shall include a link to the PR, a brief description of the proposed change, and indicate any desired timeline for approval. It is especially important that proposals tied to contractual milestones be so designated to ensure that Members are alerted to potential time pressures.

The Straw Poll Comment is used by the ASC as a mechanism for obtaining a sense of how many people have viewed the PR and are following the discussion. This serves no official purpose other than to help define participation in the consensus building process and record the tally of votes per emoji category, as defined above.

Anyone indicating concern must provide a comment on the ticket following the Straw Poll Comment describing their concern. Sufficient detail should be provided to allow the author(s) to respond to the specific concerns. The author(s) can then work to address the concern, modifying the PR as required - modification should be accompanied by an appropriate comment citing the original participant who raised the concern to ensure they are notified of the change. Participants and Members shall indicate their acceptance or rejection of the modification by so marking the change comment and adjusting their original "vote" on the Straw Poll if appropriate.

The ASC Co-Chairs shall monitor the status of discussion on the PR. Once discussion appears to have consolidated (either measured by participation in the Straw Poll, rate of comments, or a combination of both), or at the request of the proposal's author(s), the Co-Chairs will schedule the proposal for a reading at the next quarterly meeting of the ASC. The proposal shall be included in the announced agenda for the meeting along with a link to the PR and the results of the Straw Poll.

At least one proposal author must attend the quarterly meeting to present the PR and participate in discussion regarding its approval. A *Reading* of the proposal shall be conducted at the meeting with the author(s) reviewing any questions/concerns raised and how they were addressed. During the presentation, someone designated by the Co-Chairs will take notes on the PR (and/or Issue) on behalf of the presenter. After the reading in the same quarterly meeting, a formal vote of the ASC shall be held (per the above voting rules) to determine if the proposal is acceptable as a provisional item. After the vote, the Co-Chairs shall set a GitHub Label corresponding to the result - either indicating "Accepted" or "Pushed Back" based on the results of the formal vote. Accepted proposals can be committed to the next minor release version by the Release Manager in accordance with their schedule. Proposals that are pushed back for further work can be brought forward again at a later date, or can be withdrawn by the author(s) by simply closing the PR.

Process flow for Provisional proposals. In summary, the flow for proposing and gaining approval of a provisional attribute and/or API consists of the following steps:

1. Open an Issue describing the proposed change to serve as a discussion forum
2. Open a Draft PR while writing the proposal itself
3. Remove the Draft PR status when ready for discussion
4. Discuss PR - only new commits can be added
5. Label the PR "RFC" and announce to the mailing list that it is ready for formal review
6. Add the Straw Poll Comment to request the PR be read at the next quarterly meeting
7. ASC Co-Chairs schedule proposal for Reading at next quarterly meeting
8. Author(s) present proposal for review at quarterly meeting
9. Formal ASC vote taken to determine if proposal is acceptable as provisional
10. ASC Co-Chairs mark proposal as either "Accepted" or "Pushed Back"
11. Accepted proposals merged by Release Managers
12. Corresponding Issue closed

Note that a provisional proposal could be accepted in as little as one quarterly meeting.

1.3.5 Elevation to Stable Status

Provisional elements of the Standard that have been included in at least two minor releases become eligible for elevation to Stable status. The elevation process begins with an Issue requesting consideration of the change, identifying the specific elements (APIs and attributes) to be included, and a justification for Stable status. Justification should be commensurate with the magnitude of the request - e.g., elevating an API to Stable likely requires more justification than a key-value attribute. While there is no clear formula for justifying elevation, typical considerations might include the breadth of adoption (by implementers, system management vendors, and/or programming libraries) and stability of the elements themselves (i.e., time since last modification).

Note that attributes can be proposed for elevation separate from the APIs that use them. Promotion of one or more attributes independent of any API may, for example, occur when a Stable API has expressed a desire to utilize them. In such cases, the proposal should clearly identify the Stable APIs making the request and provide some justification for it.

When ready for consideration, the author(s) shall create a corresponding PR (linked to the Issue) in a Draft PR state until ready for submission for review and label the PR with the "Elevate" tag, thereby requesting that the matter be placed on the agenda of the next quarterly meeting (subject to the agenda finalization deadline). As with the process for Provisional acceptance, the Straw Poll mechanism is utilized to provide an initial screening of the proposal. Passing of the Straw Poll *does not* confer elevation to Stable status, but serves instead as a means for quickly pushing back proposals that lack significant support. This is intended to both protect the time of ASC Members and author(s), and to help guide author(s) wishing to continue pursuit of the proposal.

Upon successfully completing the Straw Poll, the Co-Chairs shall label the PR as "Eligible" to indicate that the proposal is eligible for formal consideration by the ASC. Given the significant commitment conferred by the PMIx community to Stable elements, formal approval of elevation requires that the ASC conduct a full presentation on the proposal at two consecutive quarterly meetings. The timing of the second meeting may be delayed at the request of the author(s) or ASC Members, subject to approval by the ASC. At the conclusion of the first meeting, a formal vote of the ASC shall be held (per the above voting rules) to determine if the proposal merits further consideration. The Co-Chairs shall set a GitHub label indicating the result as either "First Vote Passed" or "Pushed Back".

At the conclusion of the second meeting where the proposal is up for consideration, another formal vote of the ASC shall be held (per the above voting rules) to determine final disposition of the proposal. The Co-Chairs shall subsequently set a GitHub label indicating the result as either "Stable" (indicating that the proposed elements have been elevated to that class) or "Pushed Back". Accepted proposals can be committed to the next major release version by the Release Manager in accordance with their schedule.

Proposals that are pushed back at any point can be modified and resubmitted by the author(s). However, the updated PR must pass through the complete elevation process from the beginning to ensure full consideration of the revised request. A new PR for this purpose can be submitted for the identical elements but must be linked to the original PR in order to retain full information regarding prior concerns and suggestions.

Process flow for Elevation proposals. In summary, the flow for proposing and gaining approval for elevation of an attribute and/or API to Stable status consists of the following steps:

1. Open an Issue describing the proposed change to serve as a discussion forum
2. Open a Draft PR while writing the proposal itself
3. Remove the Draft PR status when ready for discussion
4. Discuss PR - only new commits can be added
5. Label the PR "Elevate" and announce to the mailing list that it is ready for formal review
6. Add the Straw Poll Comment to request feedback
7. If it passes the Straw Poll, then ASC Co-Chairs label it as "Eligible" for consideration at next quarterly meeting
8. ASC Co-Chairs schedule proposal for Reading at next quarterly meeting
9. Author(s) present proposal for review at quarterly meeting
10. Formal ASC vote taken to designate as "Pushed Back" or "First Vote Passed"
11. If passed, then author(s) present it again at a following quarterly meeting
12. Formal ASC vote taken to designate as "Pushed Back" or "Accepted as Stable"

13. Accepted proposals merged by Release Managers
14. Corresponding Issue closed

Note that a proposal requires a minimum of two quarterly meetings to be accepted and that actual publication of the accepted proposal must be included in the next major release of the standard.

1.3.6 Deprecation

Deprecation of one or more existing elements of the Standard, be they APIs or attributes, starts by mirroring the path of a newly proposed modification. An Issue proposing the deprecation and explaining the reasons behind it shall be opened for initial discussion. Impact on backward or cross-version compatibility must be identified in the discussion to ensure adequate consideration is given to such issues. When ready (either in parallel with the Issue or after an appropriate amount of discussion on the Issue has concluded), the author(s) must file a Deprecation PR containing the specific changes to the document where deprecation warnings must be added to the Standard and/or wording needs to be modified - examples are provided in the GitHub Deprecation Template. The PR is opened in a Draft PR state until ready for submission for review. When ready for consideration, the author(s) shall label the PR with the "Deprecate" tag, thereby requesting that the matter be placed on the agenda of the next quarterly meeting (subject to the agenda finalization deadline). The same Straw Poll voting mechanism, including presentation at a quarterly meeting, shall be used to determine if the proposal merits deeper consideration by the ASC.

Deprecation of attributes can be proposed separately from the APIs that support them - i.e., it is not required that an API which utilizes an attribute be deprecated at the same time as the attribute. The process of deprecating an attribute is, however, somewhat more complicated as it may (a) be used in multiple Stable as well as Provisional APIs and (b) impact the behavior of an API that supports it. Care must be taken, therefore, to identify *all* places in the Standard that are impacted by the deprecation proposal. Where Stable APIs are impacted, a deprecation warning for the change in *behavior* must be included along with warning of deprecation of the attribute itself.

Deprecation of Provisional elements proceeds along an accelerated path. Upon initial acceptance based on the final results of the Straw Poll, the Co-Chairs shall schedule a final formal vote on the proposed deprecation at the next quarterly meeting. Thus, a decision to deprecate an existing Provisional element requires both a Straw Poll and a single formal vote by the ASC before being accepted. The accepted deprecation proposal shall be labeled by the Co-Chairs as "DEPRECATED" and with the version of the Standard where the deprecated elements shall be removed. Provisional elements must only be marked as deprecated for a minimum of one *minor* release of the Standard before being removed - i.e., a provisional element that is marked for deprecation in release A.1 can be removed from the Standard in release A.2.

Deprecation of Stable elements follows a similar path, but requires *two* formal votes by the ASC before being accepted. This mirrors the acceptance path for elevating the elements to Stable status. The "double hurdle" is intentional as removing an existing Stable element of the Standard is considered to be a significant matter potentially impacting a wide range of the PMIx community. The affected elements shall remain in the Standard, marked for deprecation, for a minimum of one *major* release - i.e., an entry that is marked for deprecation in release A.0 cannot be removed from the Standard until at least the A+2.0 release.

Once officially deprecated, the PR carrying the deprecation warning(s) shall be committed to the Standard by the Release Manager in accordance with their schedule. Extensions to the deprecation cycle (i.e., the length of time the elements must remain in the Standard marked as "deprecated") can be granted on a case-by-case basis upon vote of the ASC.

At the conclusion of the deprecation cycle, a second "removal" PR shall be submitted by the author(s) to actually remove the deprecated element(s) from the PMIx Standard document. When ready for consideration, the author(s) shall label the PR with the "Removal" tag, thereby requesting that the matter be placed on the agenda of the next quarterly meeting (subject to the agenda finalization deadline). The Straw Poll approval process utilized for Provisional proposals is utilized to ensure adequate review is performed to avoid inadvertent removal of elements not included in the original deprecation proposal. Once the process has been concluded, the Release Manager shall commit the change in accordance with their schedule.

Process flow for Deprecation proposals. In summary, the flow for proposing and gaining approval for elevation of an attribute and/or API to Stable status consists of the following steps:

1. Open an Issue describing the proposed deprecation to serve as a discussion forum

2. Open a Draft PR while writing the proposal itself
3. Remove the Draft PR status when ready for discussion
4. Discuss PR - only new commits can be added
5. Label the PR "Deprecate" and announce to the mailing list that it is ready for formal review
6. Add the Straw Poll Comment to request feedback
7. If it passes the Straw Poll, then ASC Co-Chairs label it as "Eligible" for consideration at next quarterly meeting
8. ASC Co-Chairs schedule proposal for Reading at next quarterly meeting
9. Author(s) present proposal for review at quarterly meeting
10. Formal ASC vote taken
    - Provisional elements: designate as "Pushed Back" or "Deprecated"
    - Stable elements: designate as "Pushed Back" or "First Vote Passed"
      - If passed, then author(s) present it again at a following quarterly meeting
      - Formal ASC vote taken to designate as "Pushed Back" or "Deprecated"
11. Accepted proposals merged by Release Managers
12. Corresponding Issue closed

Note that a proposal to deprecate Provisional elements can be approved in as little as one quarterly meeting, while a proposal to deprecate Stable elements requires a minimum of two quarterly meetings to be accepted.

1.3.7 Major Text Changes

Major Text Changes are defined as significant changes to descriptive text in the document such as introductory text in a chapter, definitions of terms, re-organization of chapter contents or chapter placement, non-errata level changes to descriptions of APIs including advice to implementors/users. A major text change can target an upcoming minor or major release of the standard, but significant changes in the semantics of existing APIs shall target major releases. The definition of "significant" is determined by the members of the ASC.

The process for major text changes begins with a GitHub Draft PR while writing the proposal, optionally proceeded by a GitHub Issue for more general discussion and guidance. When the GitHub PR is ready for discussion the author(s) must remove the Draft PR status, add the "RFC" and "Major Text Change" labels, and add the Straw Poll Comment to it. Then the author(s) must send an announcement of the PR to the PMIx community mailing list requesting consideration of the PR as a major text change. This starts the more formal discussion and Straw Poll voting process.

The Straw Poll Comment is used by the ASC as a mechanism for obtaining a sense of how many people have viewed the PR and are following the discussion. This serves no official purpose other than to help define participation in the consensus building process and record the tally of votes per emoji category, as defined above. The author(s) must engage with any concerns raised on the PR in an attempt to resolve those concerns before moving to the next stage in the process.

The ASC Co-Chairs shall monitor the status of discussion on the PR. Once discussion appears to have consolidated (either measured by participation in the Straw Poll, rate of comments, or a combination of both), or at the request of the proposal's author(s), the Co-Chairs will schedule the proposal for a reading at the next quarterly meeting of the ASC. The proposal shall be included in the announced agenda for the meeting along with a link to the PR. The PR shall explicitly specify if it targets an upcoming minor or major release version of the standard. The Co-Chairs shall label the PR as "Eligible" to indicate that the proposal is eligible for formal consideration by the ASC at the next quarterly meeting.

Formal approval of a major text change requires that the ASC conduct a full presentation on the proposal at two consecutive quarterly meetings. The timing of the second meeting may be delayed at the request of the author(s) or ASC Members, subject to approval by the ASC.

At least one proposal author must attend the quarterly meeting to present the PR and participate in discussion regarding its approval. A *Reading* of the proposal shall be conducted at the meeting with the author(s) reviewing any questions/concerns raised and how they were addressed. During the presentation, someone designated by the Co-Chairs will take notes on the PR (and/or Issue) on behalf of the presenter. At the conclusion of the reading, a formal vote of the ASC shall be held (per the above voting rules) to determine if the proposal merits further consideration or pushed back for further modification. The Co-Chairs shall set a GitHub label indicating the result as either "First Vote Passed' or "Pushed Back".

At the conclusion of the second meeting where the proposal is up for consideration, another formal vote of the ASC shall be held (per the above voting rules) to determine final disposition of the proposal. The Co-Chairs shall subsequently set a GitHub label indicating the result as either "Accepted" or "Pushed Back". Accepted proposals can be committed to the next ~~major~~ specified target release version by the Release Manager in accordance with their schedule.

Proposals that are pushed back at any point can be modified and resubmitted by the author(s). However, the updated PR must pass through the complete major text change process from the beginning to ensure full consideration of the revised request. A new PR for this purpose can be submitted for the identical elements but must be linked to the original PR in order to retain full information regarding prior concerns and suggestions. Alternatively, the ASC may suggest a Revision Exception vote instead of pushing back the PR. The ASC may also suggest a Revision Exception vote to change the target release for the PR; in this case, the PR should be rebased on the new target release with no other changes to its history.

Process flow for Major Text Changes In summary, the flow for proposing and gaining approval for a major text change consists of the following steps:

1. Optionally, open an Issue describing the proposed change to serve as a discussion forum
2. Open a Draft PR while writing the proposal itself
3. When the PR is ready for discussion:
   (a) Remove the Draft PR status
   (b) Label the PR "RFC" and "Major Text Change"
   (c) Milestone the PR with the target (minor or major) release of the standard
   (d) Add the Straw Poll Comment to request feedback
   (e) Send announcement to the mailing list that it is ready for formal review
4. Discuss PR - only new commits can be added
5. ASC Co-Chairs will schedule the proposal for Reading at next quarterly meeting and label the PR with "Eligible"
6. Author(s) present proposal for review at quarterly meeting
7. Formal ASC vote taken to designate as "Pushed Back" or "First Vote Passed"
8. If passed, then author(s) present it again at a following quarterly meeting
9. Formal ASC vote taken to designate as "Pushed Back" or "Accepted"
10. Accepted proposals merged by Release Managers
11. Any corresponding Issue(s) are closed

Note that a proposal requires a minimum of two quarterly meetings to be accepted~~and that actual publication of the accepted proposal must be included in the next major release of the standard~~.

1.3.8 Errata Changes

Errata changes are defined as "small" changes that correct errors or clarify ambiguities in the document. The definition of "small" is determined by the members of the ASC. As such, any errata item may be deemed by an ASC member as a major change (e.g., because it changes semantics in a subtle, but significant way) and thus need to go through a more formal process such as the Provisional acceptance process. Note that for typos or grammar edits the Minor Document Changes Guidance applies.

The process for errata changes begins with a GitHub Draft PR while writing the proposal, optionally proceeded by a GitHub Issue for more general discussion and guidance. When the GitHub PR is ready for discussion the authors(s) must remove the Draft PR status, add the "RFC" and "Errata" labels, and add the Straw Poll Comment to it. Then the author(s) must send an announcement of the PR to the PMIx community mailing list requesting consideration of the PR as an errata change. This starts the more formal discussion and Straw Poll voting process.

The Straw Poll Comment is used by the ASC as a mechanism for obtaining a sense of how many people have viewed the PR and are following the discussion. This serves no official purpose other than to help define participation in the consensus building process and record the tally of votes per emoji category, as defined above. The author(s) must engage with any concerns raised on the PR in an attempt to resolve those concerns before moving to the next stage in the process.

The ASC Co-Chairs shall monitor the status of discussion on the PR. Once discussion appears to have consolidated (either measured by participation in the Straw Poll, rate of comments, or a combination of both), or at the request of the proposal's author(s), the Co-Chairs will schedule the proposal for a reading at the next quarterly meeting of the ASC. The proposal shall be included in the announced agenda for the meeting along with a link to the PR. The Co-Chairs shall label the PR as "Eligible" to indicate that the proposal is eligible for formal consideration by the ASC at the next quarterly meeting.

At least one proposal author must attend the quarterly meeting to present the PR and participate in discussion regarding its approval. A *Reading* of the proposal shall be conducted at the meeting with the author(s) reviewing any questions/concerns raised and how they were addressed. During the presentation, someone designated by the Co-Chairs will take notes on the PR (and/or Issue) on behalf of the presenter. At the conclusion of the reading, a formal vote of the ASC shall be held (per the above voting rules) to determine if the proposal is accepted or pushed back for further modification. The Co-Chairs shall set a GitHub label indicating the result as either "Accepted' or "Pushed Back".

Accepted proposals can be committed to the next minor release version by the Release Manager in accordance with their schedule. Proposals that are pushed back for further work can be brought forward again at a later date, or can be withdrawn by the author(s) by closing the PR.

Process Flow for Errata Changes. In summary, the flow for proposing and gaining approval of a errata change consists of the following steps:

1. Optionally, open an Issue describing the proposed change to serve as a discussion forum
2. Open a Draft PR while writing the proposal itself
3. When the PR is ready for discussion:
   (a) Remove the Draft PR status
   (b) Label the PR "RFC" and "Errata"
   (c) Add the Straw Poll Comment to request feedback
   (d) Send announcement to the mailing list that it is ready for formal review
4. Discuss PR - only new commits can be added
5. ASC Co-Chairs will schedule the proposal for Reading at next quarterly meeting and label the PR with "Eligible"
6. Author(s) present proposal for review at quarterly meeting
7. Formal ASC vote taken to designate as "Pushed Back" or "Accepted"
8. Accepted proposals merged by Release Managers
9. Any corresponding Issue(s) are closed

Note that an errata changes proposal could be accepted in as little as one quarterly meeting.