

PLEP-0001 – Purpose and Guidelines for PlasmaPy Enhancement Proposals

Nicholas A. Murphy

PLEP	1
author(s)	Nicholas A. Murphy
contact email	namurphy@cfa.harvard.edu
date created	2017-11-13
date last revised	2018-09-18
type	process
status	accepted

Introduction

PlasmaPy Enhancement Proposals (PLEPs) are design documents that provide information to the PlasmaPy community, describe decision-making processes, or propose major changes or enhancements to the code. PLEPs are intended to be the primary mechanism for proposing major changes to the direction of PlasmaPy, collecting community feedback, and documenting the reasoning behind major decisions.

Types of PLEPs

There are three primary types of PLEPs:

- A **standard PLEP** introduces and describes a major change to the PlasmaPy code base. Standard PLEPs may describe a new feature or subpackage, major changes to an existing package, or a backwards incompatible change to the application programming interface (API). A standard PLEP will start out as a proposal and eventually evolve into a design document if accepted.
- A **process PLEP** describes a new process or change to an existing process in the management and coordination of PlasmaPy. Examples include changes to PlasmaPy

decision-making processes or management structure, guidelines, or procedures. A process PLEP will start out as a proposal and eventually evolve into a document on the governance of PlasmaPy.

- An **informational PLEP** provides information and does not describe any changes.

PLEP workflow

A PLEP may come about when a contributor has an idea for how to improve PlasmaPy, or when a Coordinating Committee member or subpackage maintainer requests a PLEP to be written before a pull request is accepted. The following subsections describe the procedure for PLEPs to be created, submitted, decided upon, amended, and implemented. Anyone who abides by the code of conduct may submit a PLEP.

Creating a PLEP

Before writing a PLEP, it is generally advisable to bring up your idea on the PlasmaPy [Matrix/Gitter](#) channel and with the Coordinating Committee and/or subpackage maintainers to get initial feedback. PLEPs are not necessary to propose minor changes, which are often best [submitted as issues](#) on the [PlasmaPy GitHub repository](#).

Each PLEP should contain a clear, concise, and well-organized description of a new idea or proposal. PLEPs should generally focus on a single topic. PLEPs should be written to be understandable to general members of the PlasmaPy community rather than just core contributors and Coordinating Committee members. However, technical details about the implementation of a proposed change should be included when needed.

Standard PLEPs are generally only needed for major improvements to the code base such as significant restructuring of the code base or adoption of a new standard. PLEPs are more likely to be needed for changes that break backward compatibility, especially for stable subpackages in development releases and after the release of version 1.0.0. PLEPs are not needed for minor changes.

All new PLEPs should begin with a copy of the PLEP template contained within the PlasmaPy/PlasmaPy-PLEPs repository. This copy should be renamed to `PLEP-nnnn.rst` where **nnnn** should be the lowest PLEP number preceded by zeros that has not been tentatively or permanently assigned by the Coordinating Committee. The template contains a suggested outline that is most appropriate for a standard PLEP. Not all sections included in the template are required for every PLEP, and sometimes it is appropriate to create different sections. PLEPs are written in the [reStructuredText](#) format.

Amending or superseding a PLEP

PLEPs may be amended or superseded as PlasmaPy grows and changes.

If a topic is already covered by an existing PLEP, then it is often appropriate to propose an amendment to that PLEP instead of writing a new PLEP. Amendments should not substantially change the spirit of the PLEP. Proposed amendments should update the “date last revised” category in the header. Amendments to PLEPs go through the same process as newly proposed PLEPs.

When major changes to a PLEP are desired, then it is most appropriate to propose a new PLEP to supersede the old PLEP. The pull request to supersede a PLEP should change the “status” category in the header of the old PLEP to “superseded by PLEP N” where N is the number of the new PLEP not preceded by zeros.

When PLEPs are amended or superseded, it is generally helpful to request reviews from the authors of the original PLEP.

Minor changes to a PLEP that do not affect its meaning or intent (e.g., formatting changes, fixes to typos, updates to links, minor rewording to improve clarity, and reversing cosmic ray bit flips) do not need to go through the full review process, but do require approval by the Coordinating Committee.

Submitting a PLEP

All new PLEPs and amendments to PLEPs should be submitted as pull requests into the PlasmaPy-PLEP repository. The pull request should be created while the PLEP is being written to allow for greater transparency and community input during the writing process. In this case, the sponsor should comment on the pull request and inform the Coordinating Committee when the proposed PLEP is ready for review.

Review process

When a new PLEP or an amendment to a PLEP is submitted, the Coordinating Committee should appoint one of its members to be the PLEP’s editor. The editor is responsible for aiding the sponsor by making sure that the PLEP follows the accepted standard and facilitating communication between the sponsor and the Coordinating Committee. The editor and sponsor may be the same person.

When the sponsor and editor decide that the proposal is ready for broader discussion, then they should email the PlasmaPy list to introduce the PLEP and request community feedback. The status of the PLEP should be changed to “Discussion” at this time. Community discussion should take place through normal communication channels such as the Matrix/Gitter channel,

on the pull request on GitHub, and community meetings. Concerns that are raised should be constructive and made in accordance with the PlasmaPy code of conduct. Important points and relevant meeting minutes should be recorded as comments on the GitHub pull request so that the community discussion may be archived. During the discussion phase, the sponsor should revise the PLEP in order to address concerns raised by the community. Others may propose changes as well. The discussion should continue until a general consensus among the PlasmaPy community has been reached.

The Coordinating Committee is tasked with making the final decision on the PLEP. A PLEP shall be accepted when at least two-thirds of the Coordinating Committee votes in favor of the PLEP. The vote may be taken in the discussion of the pull request on GitHub, over email, or at a Coordinating Committee meeting where voting is recorded in the minutes.

PLEP status

The status of a standard or process PLEP may be any of the following:

- **In preparation:** The PLEP is currently being written. Community input is welcome during this phase so that concerns may be addressed earlier rather than later.
- **Discussion:** The PLEP is currently being considered and getting community feedback before a decision has been made. This is the default status for informational PLEPs that have been completed.
- **Accepted:** The PLEP has been accepted and it will be assigned a number and merged into the PlasmaPy/PlasmaPy-PLEPs repository. A decision rationale for standard PLEPs should be drafted and added to the PLEP by the sponsor of the PLEP or by someone appointed by the Coordinating Committee. Features proposed in standard PLEPs may now be implemented. Process PLEPs come into effect when accepted.
- **Implemented:** The feature discussed in a standard PLEP has been fully implemented and merged into the main repository. At least half of the Coordinating Committee must agree that the implementation (including documentation and tests) is complete. A summary of the implementation process should be added to the PLEP when this status is reached. This summary should include links to the issues and pull requests associated with this PLEP that were created after the PLEP was accepted.
- **Declined:** The community and Coordinating Committee decided against a proposed PLEP. A decision rationale should be provided by the sponsor, editor, and/or Coordinating Committee. The PLEP should still be assigned a number and merged into the main repository. A future PLEP may supersede this decision.
- **Superseded:** The PLEP is no longer in effect and has been replaced by another PLEP.

Informational PLEPs that are being written should have a status of **in preparation**, and **informational** when it is sufficiently complete. Informational PLEPs may also have a status of **outdated** when the some of the information is out-of-date and needs updating, or

obsolete when the PLEP is no longer relevant.

Archiving PLEPs

When a PLEP has been decided upon and merged into the repository, the Coordinating Committee will upload the PLEP to [Zenodo](#) for permanent archiving and so that the PLEP may get a Digital Object Identifier (DOI) and therefore be citable. The PLEP should be included as part of the [PlasmaPy Community on Zenodo](#). The title should be of the form “PlasmaPy Enhancement Proposal *number: title*”. All PLEPs on Zenodo should be versioned to allow for eventual amendments and revisions.