Abstract

This PHEP provides a template to make it easier to write a PHEP-1 compliant PHEP. It does not establish or enforce any standard. Each section contains explanatory text for how to write that section in an actual PHEP.

The majority of the text should be replaced with the text of your proposed PHEP. See Specification for further directions on using this template.

This section of your PHEP should include a short description of the issue being addressed.

Motivation

Discussions in the drafing of PHEP 1 indicated a desire to have a template to make it easier for potential PHEP writers to comply with the format, address the required elements, and avoid the "blank page" problem.

The motivation must clearly explain why the existing standards are inadequate to address the problem that the PHEP solves. This can include collecting documented support for the PHEP from important projects in the PyHC ecosystem.

Rationale

The template model should help make the initial drafting of a PHEP a fairly simple copy-paste-replace operation. Although in theory any PHEP could be taken as a template, each PHEP is unique and it is easier to have one that spans the set of expected sections, while minimizing additional information.

The rationale fleshes out the specification by describing why particular design decisions were made. It should describe alternate designs that were considered and related work.

The rationale should provide evidence of consensus within the community and discuss important objections or concerns raised during discussion.

Specification

Use this PHEP by copying it to a new file, usually named `phep-9999.md`, and editing it. (See PHEP 1 for details on the process.) The text is in Markdown (specifically CommonMark) and can be directly edited by any text editor. You can check the rendering of text fragments here, and editors with explicit Markdown support include:

- ReText
- PyCharm

Start by updating the preamble as follows:

- **PHEP** number should be 9999 until a new number is assigned by the editors.
- **Title** should be the title of your PHEP.
- **Author** should be updated with your name and (ideally) email address and ORCID.
- **Discussions-To** is the URL of your pull request once it is opened (so this usually is not updated until the PR is open).
- **Revision** is 1.
- **Status** is Draft.
- **Type** is "Standards Track" if proposing a requirement for PyHC packages; "Informational" for best practices and similar information; "Process" if proposing a means by which PyHC conducts itself (see PHEP 1).
- **Content-Type** must remain, verbatim `text/markdown; charset=UTF-8; variant=CommonMark`.
- **Requires** should list the number of all PHEPs that this one depends on (i.e., that this one does not make sense without). Leave the header out if there are none.
- **Created** is the date the PHEP was *assigned a number* by the editors.
- **Post-History** should include the date of the first push to the PR (and subsequent pushes separated by commas) in `dd-mmm-yyyy`.
- **Replaces** should be the number of a final PHEP which this PHEP proposes to replace, if appropriate (otherwise exclude this header).

Now go through each section and fill it in with the contents of your PHEP, removing the text of this one as you go. Each section usually contains two parts. First is the relevant content for this template PHEP (if necessary). This is followed by the information that your PHEP should have within that section.

The sections are not all required (in fact, strictly, none are required), but the information should be there. The author should use whatever means of organization seems appropriate, including deleting or consolidating sections, breaking sections into multiple sections, and using subsections. It is recommended to add HTML anchors after each heading to make linking easier in the future.

The technical specification should completely describe the proposed standard. The specification should be detailed enough to allow competing, interoperable implementations, where applicable.

Backwards Compatibility

As the PHEP process is new, this template PHEP introduces no compatibility concerns.

All PHEPs that introduce backwards incompatibilities must describe these incompatibilities and their severity. The PHEP must explain how the author proposes to deal with these incompatibilities. If there are none, an explicit statement that there are no backward incompatibilities is preferred to removing the section.

Security Implications

This PHEP raises no security implications as it does not interact with any executing code.

Any security concerns should be explicitly written out; as with backwards compatibility, an explicit statement of no implications is preferred to removing the section.

How to Teach This

This PHEP is, itself, meant as a teaching aid for PHEP 1. Demonstrating the process of using the template, e.g. as part of a PyHC telecon, may be appropriate.

This section may include key points and recommended documentation changes that would help users, new and experienced, apply the PHEP to their work. This section should also document any changes the PHEP makes relative to a PHEP it replaces or some other widely-used standard or reference. This allows an

implementation compliant with a previous PHEP (or other standard) to be easily updated for compliance with the PHEP.

Reference Implementation

In a sense, this PHEP is itself a reference implementation.

A reference implementation is not always necessary, but is strongly suggested for Standards Track PHEPs. In that case it would include a reference to an existing or new PyHC project which is compliant with the proposed standard and a description of how it complies. The description should point out where in the compliant project's repository the standard is implemented and how it fulfils the requirements.

The reference implementation must be completed before any PHEP is given status "Final", but it need not be completed before the PHEP is accepted. While there is merit to the approach of reaching consensus on the specification and rationale before writing code, the principle of "rough consensus and running code" is still useful when it comes to resolving many discussions of API details.

The final implementation must include test code and documentation appropriate for the relevant PyHC project(s).

Rejected Ideas

Throughout the discussion of a PHEP, various ideas will be proposed which are not incorporated. Those rejected ideas should be recorded along with the reasoning as to why they were rejected. This helps record the thought process behind the final version of the PHEP and prevents people from bringing up the same rejected idea again in subsequent discussions. The author should also solicit community input on a PHEP's idea before drafting the PHEP; substantial suggestions rejected during this process should also be captured.

This should not be a full recap of the back-and-forth discussion, but only a summary of the significant ideas which were not selected.

Open Issues

While a PHEP is in draft, ideas can come up which warrant further discussion. Those ideas should be recorded so people know that they are being thought about but do not have a concrete resolution. Suggestions may also arise from preliminary conversations before the PHEP is drafted, which should be captured if they are not resolved before writing begins. This helps make sure all issues required for the PHEP to be ready for consideration are complete.

Footnotes

A collection of footnotes cited in the PHEP, and a place to list non-inline hyperlink targets. Other notes may be included in this section as appropriate, or it may be excluded.

Revisions

This section contains a brief summary of changes by revision number, including the date of approval.

Revision 1 (13-Nov-2024): Initial approval.

Copyright

The text following this paragraph should be included verbatim. The BibTeX code should be updated with the PHEP number (once it is assigned) in both the citation key and the `number` tag, the author, title, and year. Once a DOI is reserved for the document (not the repository), the BibTeX citation should be updated with the DOI for the specific revision of the PHEP.

This document is placed in the public domain or under the CC0-1.0-Universal license, whichever is more permissive. It should be cited as:

```
@techreport(phep2,
  author = {Jonathan T. Niehof},
  title  = {PHEP Template},
    year = {2023--2024},
    type = {PHEP},
  number = {2},
     doi = {10.5281/zenodo.14187913}
)
```