

BIP: 1  
Title: BIP Purpose and Guidelines  
Author: Amir Taaki <genjix@riseup.net>  
Comments-Summary: No comments yet.  
Comments-URI: <https://github.com/bitcoin/bips/wiki/Comments:BIP-0001>  
Status: Replaced  
Type: Process  
Created: 2011-09-19  
Superseded-By: 2

## What is a BIP?

BIP stands for Bitcoin Improvement Proposal. A BIP is a design document providing information to the Bitcoin community, or describing a new feature for Bitcoin or its processes or environment. The BIP should provide a concise technical specification of the feature and a rationale for the feature.

We intend BIPs to be the primary mechanisms for proposing new features, for collecting community input on an issue, and for documenting the design decisions that have gone into Bitcoin. The BIP author is responsible for building consensus within the community and documenting dissenting opinions.

Because the BIPs are maintained as text files in a versioned repository, their revision history is the historical record of the feature proposal.

## BIP Types

There are three kinds of BIP:

- A Standards Track BIP describes any change that affects most or all Bitcoin implementations, such as a change to the network protocol, a change in block or transaction validity rules, or any change or addition that affects the interoperability of applications using Bitcoin.
- An Informational BIP describes a Bitcoin design issue, or provides general guidelines or information to the Bitcoin community, but does not propose a new feature. Informational BIPs do not necessarily represent a Bitcoin community consensus or recommendation, so users and implementors are free to ignore Informational BIPs or follow their advice.
- A Process BIP describes a process surrounding Bitcoin, or proposes a change to (or an event in) a process. Process BIPs are like Standards Track BIPs but apply to areas other than the Bitcoin protocol itself. They may propose an implementation, but not to Bitcoin's codebase; they often require community consensus; unlike Informational BIPs, they are more than recommendations, and users are typically not free to ignore them. Examples include procedures, guidelines, changes to the decision-making process, and changes to the tools or environment used in Bitcoin development. Any meta-BIP is also considered a Process BIP.

## BIP Work Flow

The BIP process begins with a new idea for Bitcoin. Each potential BIP must have a champion -- someone who writes the BIP using the style and format described below, shepherds the discussions in the appropriate forums, and attempts to build community consensus around the idea. The BIP champion (a.k.a. Author) should first attempt to ascertain whether the idea is BIP-able. Posting to the [bitcoin-dev@lists.linuxfoundation.org](mailto:bitcoin-dev@lists.linuxfoundation.org) mailing list (and maybe the Development & Technical Discussion forum) is the best way to go about this.

Vetting an idea publicly before going as far as writing a BIP is meant to save both the potential author and the wider community time. Many ideas have been brought forward for changing Bitcoin that have been rejected for various reasons. Asking the Bitcoin community first if an idea is original helps prevent too much time being spent on something that is guaranteed to be rejected based on prior discussions (searching the internet does not always do the trick). It also helps to make sure the idea is applicable to the entire community and not just the author. Just because an idea sounds good to the author does not mean it will work for most people in most areas where Bitcoin is used. Small enhancements or patches often don't need standardisation between multiple projects; these don't need a BIP and should be injected into the relevant Bitcoin development work flow with a patch submission to the applicable Bitcoin issue tracker.

Once the champion has asked the Bitcoin community as to whether an idea has any chance of acceptance, a draft BIP should be presented to the bitcoin-dev mailing list. This gives the author a chance to flesh out the draft BIP to make it properly formatted, of high quality, and to address additional concerns about the proposal. Following a discussion, the proposal should be sent to the bitcoin-dev list and the BIP editor with the draft BIP. This draft must be written in BIP style as described below, else it will be sent back without further regard until proper formatting rules are followed.

BIP authors are responsible for collecting community feedback on both the initial idea and the BIP before submitting it for review. However, wherever possible, long open-ended discussions on public mailing lists should be avoided. Strategies to keep the discussions efficient include: setting up a separate SIG mailing list for the topic, having the BIP author accept private comments in the early design phases, setting up a wiki page or git repository, etc. BIP authors should use their discretion here.

It is highly recommended that a single BIP contain a single key proposal or new idea. The more focused the BIP, the more successful it tends to be. If in doubt, split your BIP into several well-focused ones.

The BIP editors assign BIP numbers and change their status. Please send all BIP-related email to the BIP editor, which is listed under BIP Editors below. Also see BIP Editor Responsibilities & Workflow. The BIP editor reserves the right to reject BIP proposals if they appear too unfocused or too broad.

Authors MUST NOT self assign BIP numbers, but should use an alias such as "bip-johndoe-infinitebitcoins" which includes the author's name/nick and the BIP subject.

If the BIP editor approves, he will assign the BIP a number, label it as Standards Track, Informational, or Process, give it status "Draft", and add it to the BIPs git repository. The BIP editor will not unreasonably deny a BIP. Reasons for denying BIP status include duplication of effort, disregard for formatting rules, being too unfocused or too broad, being technically unsound, not providing proper motivation or addressing backwards compatibility, or not in keeping with the Bitcoin philosophy. For a BIP to be accepted it must meet certain minimum criteria. It must be a clear and complete description of the proposed enhancement. The enhancement must represent a net improvement. The proposed implementation, if applicable, must be solid and must not complicate the protocol unduly.

The BIP author may update the Draft as necessary in the git repository. Updates to drafts may also be submitted by the author as pull requests.

Standards Track BIPs consist of two parts, a design document and a reference implementation. The BIP should be reviewed and accepted before a reference implementation is begun, unless a reference implementation will aid people in studying the BIP. Standards Track BIPs must include an implementation -- in the form of code, a patch, or a URL to same -- before it can be considered Final.

Once a BIP has been accepted, the reference implementation must be completed. When the reference implementation is complete and accepted by the community, the status will be changed to "Final".

A BIP can also be assigned status "Deferred". The BIP author or editor can assign the BIP this status when no progress is being made on the BIP. Once a BIP is deferred, the BIP editor can re-assign it to draft status.

A BIP can also be "Rejected". Perhaps after all is said and done it was not a good idea. It is still important to have a record of this fact.

BIPs can also be superseded by a different BIP, rendering the original obsolete. This is intended for Informational BIPs, where version 2 of an API can replace version 1.

The possible paths of the status of BIPs are as follows:

Some Informational and Process BIPs may also have a status of "Active" if they are never meant to be completed. E.g. BIP 1 (this BIP).

## What belongs in a successful BIP?

Each BIP should have the following parts:

- Preamble -- RFC 822 style headers containing meta-data about the BIP, including the BIP number, a short descriptive title (limited to a maximum

of 44 characters), the names, and optionally the contact info for each author, etc.

- Abstract -- a short (~200 word) description of the technical issue being addressed.
- Copyright/public domain -- Each BIP must either be explicitly labelled as placed in the public domain (see this BIP as an example) or licensed under the Open Publication License.
- Specification -- The technical specification should describe the syntax and semantics of any new feature. The specification should be detailed enough to allow competing, interoperable implementations for any of the current Bitcoin platforms (Satoshi, BitcoinJ, bitcoin-js, libbitcoin).
- Motivation -- The motivation is critical for BIPs that want to change the Bitcoin protocol. It should clearly explain why the existing protocol specification is inadequate to address the problem that the BIP solves. BIP submissions without sufficient motivation may be rejected outright.
- Rationale -- The rationale fleshes out the specification by describing what motivated the design and why particular design decisions were made. It should describe alternate designs that were considered and related work, e.g. how the feature is supported in other languages.
- The rationale should provide evidence of consensus within the community and discuss important objections or concerns raised during discussion.
- Backwards Compatibility -- All BIPs that introduce backwards incompatibilities must include a section describing these incompatibilities and their severity. The BIP must explain how the author proposes to deal with these incompatibilities. BIP submissions without a sufficient backwards compatibility treatise may be rejected outright.
- Reference Implementation -- The reference implementation must be completed before any BIP is given status "Final", but it need not be completed before the BIP is accepted. It is better to finish the specification and rationale first and reach consensus on it before writing code.
- The final implementation must include test code and documentation appropriate for the Bitcoin protocol.

## BIP Formats and Templates

BIPs should be written in mediawiki or markdown format.

### BIP Header Preamble

Each BIP must begin with an RFC 822 style header preamble. The headers must appear in the following order. Headers marked with "\*" are optional and are described below. All other headers are required.

```

BIP: <BIP number>
Title: <BIP title>
Author: <list of authors' real names and optionally, email addrs>
* Discussions-To: <email address>
  Status: <Draft | Active | Accepted | Deferred | Rejected |
          Withdrawn | Final | Superseded>
  Type: <Standards Track | Informational | Process>
  Created: <date created on, in ISO 8601 (yyyy-mm-dd) format>
* Post-History: <dates of postings to bitcoin mailing list>
* Replaces: <BIP number>
* Superseded-By: <BIP number>
* Resolution: <url>

```

The Author header lists the names, and optionally the email addresses of all the authors/owners of the BIP. The format of the Author header value must be

```
Random J. User <address@dom.ain>
```

if the email address is included, and just

```
Random J. User
```

if the address is not given.

If there are multiple authors, each should be on a separate line following RFC 2822 continuation line conventions.

Note: The Resolution header is required for Standards Track BIPs only. It contains a URL that should point to an email message or other web resource where the pronouncement about the BIP is made.

While a BIP is in private discussions (usually during the initial Draft phase), a Discussions-To header will indicate the mailing list or URL where the BIP is being discussed. No Discussions-To header is necessary if the BIP is being discussed privately with the author, or on the bitcoin email mailing lists.

The Type header specifies the type of BIP: Standards Track, Informational, or Process.

The Created header records the date that the BIP was assigned a number, while Post-History is used to record the dates of when new versions of the BIP are posted to bitcoin mailing lists. Both headers should be in yyyy-mm-dd format, e.g. 2001-08-14.

BIPs may have a Requires header, indicating the BIP numbers that this BIP depends on.

BIPs may also have a Superseded-By header indicating that a BIP has been rendered obsolete by a later document; the value is the number of the BIP that replaces the current document. The newer BIP must have a Replaces header containing the number of the BIP that it rendered obsolete.

## Auxiliary Files

BIPs may include auxiliary files such as diagrams. Image files should be included in a subdirectory for that BIP. Auxiliary files must be named BIP-XXXX-Y.ext, where "XXXX" is the BIP number, "Y" is a serial number (starting at 1), and "ext" is replaced by the actual file extension (e.g. "png").

## Transferring BIP Ownership

It occasionally becomes necessary to transfer ownership of BIPs to a new champion. In general, we'd like to retain the original author as a co-author of the transferred BIP, but that's really up to the original author. A good reason to transfer ownership is because the original author no longer has the time or interest in updating it or following through with the BIP process, or has fallen off the face of the 'net (i.e. is unreachable or not responding to email). A bad reason to transfer ownership is because you don't agree with the direction of the BIP. We try to build consensus around a BIP, but if that's not possible, you can always submit a competing BIP.

If you are interested in assuming ownership of a BIP, send a message asking to take over, addressed to both the original author and the BIP editor. If the original author doesn't respond to email in a timely manner, the BIP editor will make a unilateral decision (it's not like such decisions can't be reversed :).

## BIP Editors

The current BIP editor is Luke Dashjr who can be contacted at [luke\\_bipeditor@dashjr.org](mailto:luke_bipeditor@dashjr.org).

## BIP Editor Responsibilities & Workflow

The BIP editor subscribes to the Bitcoin development mailing list. All BIP-related correspondence should be sent (or CC'd) to [luke\\_bipeditor@dashjr.org](mailto:luke_bipeditor@dashjr.org).

For each new BIP that comes in an editor does the following:

- Read the BIP to check if it is ready: sound and complete. The ideas must make technical sense, even if they don't seem likely to be accepted.
- The title should accurately describe the content.
- Edit the BIP for language (spelling, grammar, sentence structure, etc.), markup (for reST BIPs), code style (examples should match BIP 8 & 7).

If the BIP isn't ready, the editor will send it back to the author for revision, with specific instructions.

Once the BIP is ready for the repository it should be submitted as a "pull request" to the bitcoin/bips repository on GitHub where it may get further feedback.

The BIP editor will:

- Assign a BIP number (almost always just the next available number, but sometimes it's a special/joke number, like 666 or 3141) in the pull request comments.
- Merge the pull request when the author is ready (allowing some time for further peer review).
- List the BIP in README.mediawiki
- Send email back to the BIP author with next steps (post to bitcoin-dev mailing list).

The BIP editors are intended to fulfill administrative and editorial responsibilities. The BIP editors monitor BIP changes, and correct any structure, grammar, spelling, or markup mistakes we see.

## History

This document was derived heavily from Python's PEP-0001. In many places text was simply copied and modified. Although the PEP-0001 text was written by Barry Warsaw, Jeremy Hylton, and David Goodger, they are not responsible for its use in the Bitcoin Improvement Process, and should not be bothered with technical questions specific to Bitcoin or the BIP process. Please direct all comments to the BIP editors or the Bitcoin development mailing list.

## Changelog

10 Oct 2015 - Added clarifications about submission process and BIP number assignment.

01 Jan 2016 - Clarified early stages of BIP idea championing, collecting community feedback, etc.