Where to commit

Once you have read up on how to commit code, a very important decision is **where to commit**? Stable-LTS, Stable, Dev (~~trunk on SVN~~ / master on Git), or experimental? You may want to checkout the Versions page. See also: Freeze and Slush and Git and SVN Combined Workflow

Dashboard

Quick overview intended to help to know **where to commit to right now** in most situations. See Branches section below for more detailed explanations.

Current status

- **All commits need to go in master first** and then be backported/cherry-picked to the branches required.
  - For example if you want to make a fix to 18.x LTS, please commit the fix to master first then backport to 24.x, 23.x, then 21.x and finally to 18.x. For more details, see the table below.
- **Upcoming releases**:
  - *24.1, 23.2, 21.6 LTS, 18.9 LTS will be the next versions released.*

Commit status and order for each open branch:

I want to commit to:	What is allowed:	Commit first to:
master (future 25x)	Functional enhancements, new features, fixes and translations	
24.x	Bug fixes and minor, self-contained enhancements	master
23.x	bug fixes and minor, self-contained enhancements	master, 24.x
21.x	Security fixes only LTS	master, 24.x, 23.x
18.x	Security fixes only LTS	master, 24.x, 23.x, 21.x

The other branches are closed, and no releases are planned. So there is no point in committing there.

Branches Again

Root: https://gitlab.com/tikiwiki/tiki or https://svn.code.sf.net/p/tikiwiki/code/

Name	*BRANCH*	*Comments*
Dev	trunk/master(will become 25.x branch in future)	Most development (new features) happens here. New features, need to be functional, but don't need to be complete. In theory, should be releasable at any time. This is the place for refactoring. Cosmetic code changes should be done here. Also: Update language strings. If you commit to master, and after you want to commit to a stable branch, please see how to merge a commit from master.
Next **Stable LTS**	branches/24.x	**Fixes and Translations only until 24.0 is released** cherry-picked from master/main/trunk
Stable STS	branches/23.x	**Minor safe enhancements, fixes and translations** cherry-picked from 24.x.
Stable LTS	branches/21.x	**Minor safe enhancements, fixes and translations** backported from 23.x.
Previous Stable LTS	branches/18.x	**Fixes only** backported from 21.x
Experimental branches	Now happening on GitLab forks	All developments for things that are not stable enough yet or just intended as proof of concept before the real work starts. These branches will never become a released branch directly, the author of the branch should create a Merge Request when it's ready (or a WIP MR before it's ready if you like.

Legend:

- STS: Standard Term Support
- LTS: Long Term Support

See Version lifecycle

Definition of "security-only" phase

- The "security-only" of the LTS period is intended for security fixes, but could include a few bug fixes as well.
  - We will review security vulnerabilities reported to the Security Team
  - Publish a fix or a way to deactivate the feature.
    - If the included code doesn't have a patch for that version
- What if a security vulnerability requires major code changes, that are not suitable for LTS?
  - We'll disable the feature via System Configuration so you can can choose to use it knowing the risks, decide not to use it, or upgrade.
- The documentation at doc.tiki.org is kept up to date for more recent versions, so expect to see there some documentation about features not available in your Tiki.

Other notes

- Commits to LTS must have been developed and tested previously on higher branches (at least master) unless they do not apply there (for example, a fix to a feature that was removed later). See here for more info.
- The community will handle merges from stable to dev, with help from a merging script during Semi-automatic merging period
- On stable branches, **try to avoid any changes to the database as this complicates things and increases possibilities for errors.** If you must, please see: Database Schema Upgrade
- If you must change the English version (but are not changing the meaning and so the translations are still valid, please use Mass spelling correction. If you can't use that, just add to Pending text corrections
- If we are close to a release, and you have a change with a risk of regression, try to consult the release manager.
- There are some things that are black and white and there are many shades of gray. In case of doubt, ask on the Dev Mailing List
- https://trunkbaseddevelopment.com/

Related

- Commit Code
- Commit Tags
- Commit Guidelines
- Git Workflow

Legacy

There are no more planned releases of versions prior to 14, as well as versions 16, 17 and 19, 20 and 22. If you are running one and commit a fix , please merge manually your fix to the appropriate branch.

Examples

[+]