

# Freezes and recommended contributions

## Suggested contributions

Consider taking an issue marked [E-help wanted](#): assign it to yourself and have a go. Feel free to ask for help in the development channel. The Lix team wants these issues fixed, but they are not high on our agenda to fix ourselves.

When in doubt, please ask the Lix team before beginning work, to make sure it is in line with our current priorities.

## New builtins

One thing that is a little bit tricky and can get contributions canned late in the process is new builtins. We should write up a more full document on this, but the gist is that the API of any builtins needs to stand the test of time as they have pretty strong compatibility and usefulness expectations.

As such, it would be greatly appreciated if work on builtins starts with an issue on `lix-project/lix` tagged RFD discussing the use case for the builtin, then, once there is rough agreement on the use case, write a simple document giving examples and design of the proposed API. Then, we can gather feedback before too much time is put into implementations that might not see the light of day.

## I don't wanna write C++. How can I help?

This is totally reasonable, C++ is not super fun, but this isn't to say there isn't help if you do choose to. Nonetheless, there are still many issues on our tracker that do not require writing any C++ that would help out immensely.

Here's some possible places to look:

- [docs issues on lix-project/lix](#). These generally involve writing documentation prose.
- [testing issues on lix-project/lix](#). A large number of these are just writing shell scripts, pytest scripts (`tests/functional2` in lix), or fixing NixOS tests
- [external project issues](#). This is a veritable grab bag of issues in external projects that we would love to have fixed. There's some in Gerrit (Java), some in Forgejo (Go), some in

meson (python), some in nixpkgs. It's unlikely these will get done without some help so we really appreciate help!

- [website issues](#) our homepage still need some love in various areas. It uses Hugo as a static site generator, so contributing requires little more than some basic HTML and CSS skills.

# Freezes

This document describes the state of freeze that Lix is in.

We *do* expect to have `main` always be in a state to run on machines you care about, unconditionally. Nightly builds should not be a problem to run in production in any freeze state.

The purpose of this policy is to set expectations of what we are looking for in contributions, rather than to set hard rules.

“ Lix is currently in status *"milkshake"*.

## ice cube

No major features or code changes are accepted that touch the core (e.g. the hot paths of the evaluator, the store), absent a good justification, good test coverage and a strong belief that they will not cause regressions. In this state, we don't recommend external contributors do substantive work outside the roadmap without speaking with the Lix team first. However, this is a guideline, and such work can be done if discussed and planned carefully beforehand.

Simple surface features with low impact are likely to be accepted with tests, assuming that they do not impact reliability.

New tests are gleefully accepted.

Bug fixes (with tests) are gleefully accepted.

For example, the following would require discussion with the Lix team before work begins, as it is likely to not fit our goals:

- Adding new features to ca-derivations
- Doing substantial not-obviously-correct refactoring to the evaluator or daemon

For example, the following would likely be accepted assuming it has tests, without needing prior discussion:

- Backports of CLI UX features from CppNix
- New UX features in the REPL or in output of other commands considered to not have stable output
- Backtrace improvements that don't touch hot paths
- Bug fixes (to non load bearing bugs; be careful around evaluator semantics!)
- Improvements to development UX

## hard ice cream

Changes that improve maintainability of the core are accepted, with careful review depending on their significance. Changes that add more complexity to the core need to pass scrutiny.

Features at the edge are accepted if they have low impact, assuming that they have tests.

## soft serve

FIXME

## milkshake

All kinds of changes are acceptable, but we still strive to keep `main` always as stable as possible and a safe decision to daily-drive for all your nixy needs. Please don't jam the ice cream machine!

---

Revision #9

Created 25 March 2024 04:35:50 by jade

Updated 27 June 2025 19:32:18 by jade