

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.

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 *"ice cube"* until public release.

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

FIXME

Revision #5

Created 25 March 2024 04:35:50 by jade

Updated 17 April 2024 07:53:40 by jade