



Process High-Impact clang-tidy and Clang Static Analyzer Requests

Peiqi Li

CppAlliance Fellowship 2026

Midterm Presentation

Mentors: Vassil Vassilev, Aaron Jomy, Martin Vassilev

Project Context && Goals

The clang-tidy and Clang Static Analyzer trackers contain a massive backlog of high-impact reports—primarily false positives, false negatives, and unsafe fix-it hints.

The Goal: Systematically triage and resolve these issues with conservative fixes driven by minimal reproducers.

Current Progress

Merged: PR #141394 (Changed IgnoreExternC default to true to protect mixed-language headers).

Merged: PR #199905 (Fixed use-after-move partial forward bug by unwrapping AST arguments).

Merged: PR #198085 (Fixed redundant-expression type alias false positives).

Pending: PR #202575 (Wiring .clang-tidy LineFilter natively into clangd).

Pending: PR #208682 (Decoupling mergeFixits() logic for clangd reuse).

Current Focus: Bridging clang-tidy and clangd

- Most developers consume clang-tidy diagnostic via Language Servers.
- clangd operates an isolated diagnostic pipeline, frequently bypassing native clang-tidy infrastructure.
- **Objective:** Decouple and integrate core clang-tidy features into clangd without breaking component boundaries.

○ Add support for .clang-tidy files to support the --line-filter option clang-tidy enhancement

#59263 · daantimmer opened on Nov 30, 2022

🔗 1 💬 12 🏆 5

○ Factor out mergeFixits() logic from DiagnosticRenderer and reuse it in clangd clang-tidy clangd

Task #207618 · HighCommander4 opened 1w ago

🔗 1 💬 2

○ [clangd] Cannot use clang-tidy's --experimental-custom-checks flag clang-tidy clangd enhancement

Feature #206696 · Togira123 opened 2w ago

💬 8

Pending PR #202575: Local LineFilter Integration

Issue: clangd users cannot filter specific macros (e.g., GTest) because .clang-tidy files lacked LineFilter support.

Fix: Exposed the option via YAML and evaluated the filter natively inside clangd's ParsedAST.cpp



```
// Inside ParsedAST.cpp:
if (IsClangTidyDiag) {
    if (Info.hasSourceManager() &&
        !passesTidyLineFilter(ClangTidyOpts, Info.getLocation(), Info.getSourceManager()))
        return DiagnosticsEngine::Ignored; // Intercepted before LSP emission
```

Pending PR #208682: Reusing mergeFixits in clangd

Issue: Applying certain clang-tidy fix-its in clangd causes syntax errors (e.g., `return (0);` -> `return0;`) due to unintended token merging.

Fix: Decoupled the logic into the `clang::edit` namespaces as a reusable utility, and wired it directly into clangd's diagnostic pipeline.

Future Work

- Land the pending clangd integrations (LineFilter and mergeFixits) and solve issue #206696.
- **Pivot to Clang Static Analyzer:** Address path-sensitive coverage gaps and performance bottlenecks (e.g., Exploded Graph path explosion) in core checkers.

Thank You!