

Roadmap for CLion 2019.3

We take application performance and code quality very seriously. Following the internal performance week / hackathon that our team held together with the IntelliJ Platform team this June, we are now planning a special Quality-targeted Release. Here's what that means in simple words:

1. We'll work to flesh out and implement the fresh ideas and fixes we tried during our performance hackathon.
2. We plan to work intensively on various performance boosts, including some massive overhauls we started earlier this year. You can expect a series of blog posts covering the progress and explaining the underlying ideas, with some measurements on referenced projects so that you can compare them with your cases.
3. We plan to focus on fixing issues and eliminating pain-points in different areas, rather than introducing new functionality. (Don't forget to [upvote](#) the pain-points that affect you the most, so that we can prioritize them to help as many users as possible!)
4. We still plan to continue our work in the directions we feel are important, such as covering Makefiles support and some others. Please read on for the details.

The following is a detailed plan for the next release cycle.

- C++ language support
 -  Mostly bug-fixing and performance improvements as mentioned above.
 -  Rework an action to switch header/source ([CPP-12920](#)).
 -  Deeper integration with the Clangd-based engine, especially in the areas where it helps eliminate performance issues and lags (for example, Clangd-based code completion).
 -  Investigation and fixes for various crashes and memory leaks in Clangd-based engine.
- Project model
 -  Built-in Makefiles support (to substitute for the current flow of [managing Makefiles projects](#)). - still under development
 -  CMake defaults for new projects ([CPP-1887](#)).
 -  Use CMake File API ([CPP-8238](#)) to allow using Ninja and other generators ([CPP-2659](#)).
- Remote development
 -  Investigate WSL v2 support opportunities ([CPP-16543](#)).
 -  Remote debugging with gdbserver via ssh ([CPP-7050](#)).
 -  Performance improvements for remote mode.
- Debugger
 -  Improve the quality of the experimental debugger for Microsoft Visual C++ toolchain (you can expect some NatVis related fixes in 2019.2.x updates already).
 -  Support for .gdbinit/.lldbinit located in project folders.
 - ~~Input/output redirection ([CPP-3153](#)).~~
 - ~~Performance investigations and improvements.~~
- Embedded Development
 -  Mostly bug-fixing (which means your feedback on the recently added functionality will be very important!).
 - ~~Console for GDBServer ([CPP-15392](#), [CPP-7103](#)).~~
- Code coverage
 -  llvm-cov/gcov integration (similar to what was recently [added in AppCode](#)).