

Takeaway after the {absl => std}::optional rename

Arthur Sonzogni

Feb 27, 2024

K. Moon asked:

Would be interested to hear more about how the experience of making this change went, and what could be improved for next time.

Clang-format:

I relied on the new "[IncludeBlocks: Regroup](#)" option to reorder the includes correctly. This was key.

Earlier CLs didn't have this and follow-up work was necessary.

I observed [@Takuto Ikuta](#) and [@Nico Weber](#) tried to bring this option by default in `.clang-format`, but unfortunately, it got [reverted](#). => Pursuing this path would help next time.

<windows>: Windows's header needs to be ordered in a specific order ;-(, causing compilation errors. I had to submit [several patches](#) to avoid clang-format to reorder them. I believe this will be resolved progressively, as the developers encounter the problem.

Status: The [new clang format](#) has shipped.

Git-blame-ignore:

After touching 150k lines, you start getting many bugs wrongly assigned to you (~5-15 every week). This slows you down, and you become the bottleneck for triaging crashes.

See [internal thread](#). TLDR: For now you can use ``git-blame-ignore`` ([example CL](#)) to make your patch less visible to bots.

=> Ideally, we would improve the bots to take the CL description into account. For instance with a *#cleanup* flag. This would avoid the need to write a second patch, and avoids the delay in between the two.

There are also many bugs filled by humans. This can't be fixed easily. A good CL title/description probably helps a bit at the margin.

Merge Conflicts:

Chrome's frequent commits (every 4-5 minutes) lead to merge conflicts for very large patches, particularly those involving 20k files. Splitting into smaller patches was needed. The sweet spot is near 1k files. This gives about 2-4 hours on average before getting a merge conflict.

We could have considered closing the tree, at the cost of slowing down several hundreds of engineers instead of me. I did not consider this option, because there were already too many unresolved issues requiring manual fixes. I preferred to iterate and submit the manual fixes progressively, along the way.

Include-What-You-Use (IWYU) Issues.

Dividing large patches leads to errors from missing transitive includes. This requires careful sequencing of changes, and adding manual fixes to unblock some patches. Many files are missing the *#include* for optional, but also for string, variants, etc... All of them were provided transitively by abseil.

=> Easier said than done: using IWYU in Chrome to automatically fix missing includes would help.