

Generating good error messages from code analysis tools is harder than it looks

 devblogs.microsoft.com/oldnewthing/20190704-00

July 4, 2019



Raymond Chen

I was writing a tool to analyze Windows Runtime winmd files, and the tool had to generate error messages when it discovered something that violated our team’s internal rules.

It turns out that generating good error messages from code analysis tools is hard. My tool detected, for example, that the name of a method parameter was improperly changed. (Method parameter names are used by some language projections, so changing the name of the parameter is a breaking change.)

It was hard enough detecting that a change occurred that violated our team’s internal rules. Now came the problem of reporting it.

My original code returned a Boolean value that said whether everything was okay, but it’s obviously a bad idea for the tool to simply say “Sorry, you made a mistake. Now go fix it.” I had to modify my code to report enough information in order to diagnose the problem and fix it. This means that instead of just “Sorry, you made a mistake,” the error message was something like “Disallowed change of parameter name from ‘index’ to ‘i’ in parameter 1 of method GetAt of interface IVector (with T = string) implemented by class MyVector in file Sample.winmd, compared to Baseline.winmd.”

I developed an appreciation for how hard it is to generate good error messages, especially if the thing you’re reporting is not something that a human being wrote directly, which means that you can’t use line numbers as reference points.

[Raymond Chen](#)

Follow

