

Production customers care a lot about hotfixes, since they may address issues that fix crashes on specific devices or platforms. We want customers to adopt hotfix releases quickly, and it's therefore important that they understand what's in them and what risks they may accept by adopting them.

As a result, it's important that Flutter customers can quickly scan through our hotfixes and identify whether they're relevant to them. We can do that by making sure that each hotfix is described carefully.

Our goal is that each hotfix summary:

1. Should be clear to a non-contributor who is developing with Flutter
2. Should be succinct (one line) for ease of scanning
3. Within above, should identify as best as possible:
 - What the scenario is (describe the problem)
 - What platforms it affects
 - In what context it may occur
 - How likely it is to occur

The goal is not to exhaustively document the issue, but to provide enough information that an educated user can quickly scan and determine whether they need to read through the bug itself to understand its applicability to their scenario.

A good approach for hotfix messages is to describe the problem in terms of the current state prior to the fix. For example:

“When \$scenario [on \$platform], \$problem_description”

Some good existing examples of hotfix messages that adopt this kind of formula:

- flutter/90783: In rare circumstances, engine may crash during app termination on iOS and macOS
- flutter/77251: Flutter may show multiple snackbars when Scaffold is nested
- flutter/98155: App crashes after upgrading to 2.10.x using webview + video_player plugin

Some harmless examples that are harder for customers to consume:

- flutter/97679: Don't remove overlay views when the rasterizer is being torn down. *Problem:* It's unclear how this affects customer apps and in which contexts *Perhaps better:* “App may crash when navigating away from embedded native platform content”
- flutter/95711: Linux builds default to building GLFW. *Problem:* Unclear why this is a problem or how this impacts customers *Perhaps better:* “On Linux, dependency on X11 breaks Wayland and embedded builds”
- flutter/97086: Windows: Fail to launch app in debug mode *Problem:* While this explains the platform and scenario, it doesn't mention the specific regression *Perhaps better:* “Flutter apps fail to launch in debug mode when compiled with Visual Studio 17.1.0”