Filing Radars

> Disclaimer: I have not worked for Apple, either in present or previously. What is posted here is based on relayed experiences and recommendations as to how to file better reports. The handling of radars can vary significantly between teams. What is written here is meant to communicate and understand of how the system works from the ouside.

> Note: This is a growing document, strategies included in the original version are subject to change. Included suggestions are only based on experiences and suggested methods to get better results in the reporting system, not guarantees in resolved radars.

What the hell is a "radar" and why should I care? "Radar" is the name of the system used by apple for filing bug reports. For us non-Apple employees, it is reachable on the web: bugreport.apple.com to file our grievances, feature requests, and bugs with Apple's products. This is known as "radar" due to the custom url-scheme used to navigate to them from inside Apple (rdar:// ######## or rdar://problem/########). Many people will tell you that filing radars is a thankless and often fruitless endeavor. More often than not your reports will be marked as a duplicate or left open without comment for long periods of time. However, there are two tricks to filing successful radars:

1. Know at least one person at Apple that you are friendly with and doesn't mind you bugging the shit out of them.
2. Know how to write a targeted bug report.

####Know Someone

No, seriously. Having someone that is willing to help you dispatch your radar off to the right group or keep track of the status when you hit dead ends is more help than I have words to describe. Bonus points if they are good enough friends to tolerate your nagging about each bug you file. If you know more than one person that will do this for you, then you are set for life. (Ask if they can radar stalk[1] you)

####Writing Targeted Reports

However, most of us don't have personal contacts at Apple that can take the time out of their day to help us with radars all the time. The only thing we can

do is try to file really good reports. Below, I have included the template and instructions displayed on the bug reporter site as to how reports should be filed.

```
Product:
Product type

Version:
Version and build number

Classification:
Select a type of problem

Reproducibility:
Select how often this problem occurs

Title:
Enter a clear and brief title that expresses the problem

Summary:
Provide a descriptive summary of the issue.

Steps to Reproduce:
In numbered format, detail the exact steps taken to produce the bug.

Expected Results:
Describe what you expected to happen when you executed the steps above.

Actual Results:
Explain what actually occurred when steps above were executed.

Regression:
Describe circumstances where the problem occurs or does not occur, such

Configuration:
Describe the circumstances where this does or does not occur (describe

Notes:
Provide additional information, such as references to related problems,
```

This seems like a pretty straight-forward form to fill out and submit. However the problem of "how do i get my radars resolved" still remains. To do this I'm going to walk through a scenario from finding a bug to filing a radar report.

####Step 1: Determining the Bug

Tracking down the actual issue will vary depending on the scenario you face. Sometimes it is hard to separate out and isolate behavior, personally I recommend starting to build any part of a project outside of the constraints of your existing code. It is less tempted to tape components together to get the

feature off the ground before refining the code. I think that this makes your code cleaner and makes reproducing bugs easier. Once you have located the issue in your own project, take it to the reproduction step.

Step 2: Repro

First thing to do when you think you have found a bug in your existing code is to create a new sample project. Open a new project in Xcode and setup a completely bare-bones environment to build in. This is your bug reproduction sample project, you will probably be asked for one of these when filing bugs so it is handy to do it right away so you don't have to come back to it in a week or two and forget how to reproduce it.

Always document every step of the way when reproducing bugs, it will save everyone time in the long run. Tools for documenting bugs properly include:

- Screenshots
- Recorded video of steps trigger the bug
- Sample code projects
- Sample files
- Results from multiple OS versions (Regression)
- Hardware configurations

The list goes on, but the idea here is to make sure you are documenting the process and replicating it exactly. This leads to a better understanding of the problem beyond being able to say "this is broken, and not how I expect it to behave". By documenting the process it can reveal not only more insight into the bug but also if there are more bugs with whatever it broken.

On complicated bugs this step is extremely vital to get anything resolved or worth the time to investigate.

Step 3: Report

When filing a radar, the form that you fill out is pretty explicit (See the template posted above). Specific and detailed radars are useful radars, so are concisely named radars. When a radar gets submitted it is unassigned. The process of screening and assignment takes place (arguably the most frustrating part of filing radars) and hopefully your radar ends up in the hands of an engineer that can take action on it.

Naming radars helps with categorization and screening. Calling a radar "syncdefaultsd crash" says what it is, however it doesn't describe the problem

as well as "10.10 (build#) | syncdefaultsd | crash on invalid objc_msgsend". Tips for naming tags:

- Include the topic of the radar. For example, filing a report about a security issue should start with "Product Security", language bugs "Objective-C" or "Swift", a specific application name and version number "Safari 7.0.3", etc.
- If the first tag is a broad subject, such a "Swift", mark a secondary tag about what aspect of swift "Standard Libary", "Compiler", "Bridge", etc. This should help narrow down the assignment of the report from the team level to the individual engineers.
- I would probably recommend 2 or 3 tags at most in a radar name. Also keep the title of the radar itself concise, most of the description should be handled by the tags.

You don't have to write novels in each field, but be explicit with your repro steps. While you may have burned hours, days, even weeks on a problem; the person reading this will have not may not have even spent 5 minutes on your particular issue. It is easy to jump to conclusions when writing up a bug report. I'd recommend explaining it to someone else on your team first to have them attempt to reproduce the problem. If they can reproduce it successfully and draw the same conclusions as you have, then your report should be able to communicate that information in the same way.

Attach any relevant information in a zip file. There is always some information that can be added to a radar beyond code. Attach a spin-dump, system trace, leak analysis, anything that might show where the problem you are experiencing is coming from. Nobody is in the business of shipping buggy software, so odds are in favor of your environment or configuration attributing or exacerbating an existing condition.

####Step 4: Feedback

As previously mentioned, radar is a closed system. If you aren't an Apple employee then you aren't able to see everything that goes on behind the scenes of your radars. Radars that sit weeks without comment probably are just comments that aren't visible to you. When posting back to radars, employees have to go through ADC to post feedback. ADC is responsible for relaying information between the engineer and originator, and ensuring NDAs are complied with. This can take some time to process. The resulting delay can give the false sense that nobody cares about your radars, or that they are mercilessly marked as a duplicate, closed, and ignored. If your radar does sit for weeks without response, you are allowed to post on them asking for a status update.

When getting feedback requests for more information, just do it. It might seem incredibly silly at times to perform request, such as take a screen recording of behavior with entering a specific URL into the address bar of Safari. Again, odds are that the engineer on the other end isn't trying to have a laugh at you. Sometimes issues can arise out of seemingly irrelevant sources. Take the time to give the information they ask for.

Sometimes you won't like the response you get back about filed reports. Asking for more clarification or readdressing the issue that you believe is the problem is acceptable. Bear in mind that "radar is forever". Anything put there stays there, there is no magic delete button for it. Being abusive or confrontational isn't going to make your case. Snark might get a laugh but also isn't terribly useful when you have frustration with the system. The worst thing that can happen is that you will be told that the issue will be closed and possibly readdressed in the future. Best case scenario is that the issue is re-examined and addressed in more detail than before.

Only being able to see your own radars is a bit unfulfilling in the overall resolution process. Some groups inside Apple use the radar system as a means of determining bug priority. This is helpful when trying to eliminate the issues impacting the largest portion of the developer community (both internal and external). Filing radars as duplicates of existing radars can carry weight and different reports on the same issue can give new details in how the bug can be reproduced and patched correctly.

####Step 5: Next Steps

Since radar reporting is a closed system, it is sometimes helpful to publish bugs in an open forum so that other people can comment on them. One such system is OpenRadar, where you can publicly file the same report and include the radar number if someone else wants to duplicate the same bug in the radar system without having to refile again. Another way is to post the radar title and number to twitter. Letting your friends know that there might be a bug that they could run into or a discrete way for people at Apple to know if they follow you.

In the process of filing a bug you are probably going to have to implement a work-around to the problem at hand. Including this in a follow-up comment on the radar can help draw attention to the desired expected behavior. This is also helpful to share incase work-arounds must be put in place by other developers that run into the same bug.

Since this is the only means of giving official feedback to Apple on any subject it really needs to be taken advantage of as much as possible (See my post about HealthKit). Someone will always have to read your report. Screening radars is a

pretty thankless job, so making it easier for them makes your issues get resolved faster.

Additions:

- Check out Apple's resources on this for [OS X](#) and [iOS](#) (Thank you Greg Parker)

///

If this blog post was helpful to you, please consider donating to keep this blog alive, thank you!

[donate to support this blog](#)

///

1. radar "stalking" is when someone subscribes to radars filed by your apple id. As a result when you file a radar they get notified of it, this is helpful for stepping around the screening process and help your reports on to the right people.[↩]