# Outside the Scope of Our Support

Published on October 18, 2026

A few days ago I wrote about reporting a bug to Google Disability Support. It's a bug that kills touch exploration for every screen reader on Android until you reboot the phone. In return I was asked to turn TalkBack off and on again, clear the cache of an app I don't use, and reboot, which is the one thing I'd already told them was the only workaround. If you haven't read that one, Have You Tried Turning TalkBack Off and On Again? has the whole story, and this post won't make much sense without it.

There's been more since. I'm going to go through it, because the ending is better than anything I could have made up. But I can tell you how I feel about it in four words, so I'll get them out of the way now.

Quite simply, fuck you.

## Where we left it

When I published the first post, I'd answered every question Google's support team asked. I'd done every troubleshooting step they gave me, including the ones that couldn't possibly work, and reported the results. They'd said that once I'd done that, the case could be forwarded to their internal team for further review. So I did it, and I waited to be forwarded.

I didn't only wait. The bug happened again the next morning, and this time I was ready for it. I took a full bug report while the phone was stuck. I captured every log buffer, before and after. I worked out exactly why the fake finger gets left on the screen: TouchExplorer gets itself into a dragging state with no valid pointer, tries to build an up event with no pointers in it, MotionEvent throws, the exception gets swallowed, and the up event that would have lifted the finger is never sent. I worked out why nothing short of a reboot clears it: the input dispatcher keeps the record of injected pointers in a verifier that is only ever thrown away when a display is removed, so turning accessibility off, turning every accessibility service off, even tearing down the whole input filter does nothing. I proved that on the phone. I found a way to clear it from the shell with a single cancel event, and a way for a screen reader to clear it with no special permissions at all. And then I wrote fixes, one patch for the accessibility code in frameworks/base and one for the input dispatcher in frameworks/native, and sent a summary of all of it back to the case, along with a request, again, for some way to get the files to an engineer, because their form still wouldn't take anything that wasn't a screenshot.

That's a root cause, a recovery method, two patches and a reproduction you can trigger on demand, all done for free, on my own time, on a phone I could only half use.

## The second agent

The next reply came from a different agent, stepping in for the first. And I'll be fair here, because I want to be: it was the best reply I'd had. It read like someone had actually read what I'd sent. It summarised the problem correctly, in proper terms: TouchExplorer failing to release injected pointers after an invalid dragging pointer state, with InputDispatcher rejecting the hover events that follow. It said it recognised the impact on screen reader users. It said it didn't want to give me an unverified technical answer to my three questions, and that they needed the team responsible for those framework components. It said, finally, that it understood the troubleshooting steps had been exhausted.

What it couldn't say was whether anyone on an engineering team would look at any of it. It didn't have an upload link, or an engineering contact. It told me to hang on to my files in case a way to send them ever turned up in the case. It was polite, honest about what it didn't know, and it went nowhere.

So I stopped waiting, and filed the bug myself.

## 102 minutes

Android has a public issue tracker. Anyone can file on it, and it has an Android Accessibility component. I wrote up everything: device and build, the natural trigger and a reliable reproduction from the shell, what happens in the logs during the trigger, why only a reboot recovers, the on-device test results, the workaround a screen reader can use, and both patches, with the reproduction helper attached. I filed it at 08:17 on a Friday morning, and sent the link to the support case so they could attach it and pass it on.

At 09:59 it was assigned to a Google engineer, with a comment saying it had been passed to the development team.

That's an hour and forty-two minutes. Disability Support had had the case for four days by then, and in four days it hadn't got as far as an engineer. The tracker did it before I'd finished my second coffee.

It's still moving. At the time of writing it's marked as assigned, severity S2, priority P2. It's on five hotlists. 114 people have given it a +1, and one of them commented that this is something that has "continued to plague users", and that it's been reported on forums like Blind Android Users. So it isn't just my phone and it isn't just me. Blind people have been hitting this for a long time, rebooting their phones, and getting nowhere, because the route Google gives them goes nowhere.

## Outside our scope

Then, late on Saturday night, Disability Support replied for the last time.

They'd reviewed my case with their internal team, it said, and the team had clarified that Google Disability Support "primarily assists with troubleshooting existing accessibility features and assistive technologies within Google products, including unexpected behavior or broken functionality."

Read that sentence again, because it's a perfect description of what I reported. Touch exploration is an existing accessibility feature. It's built into Android, which is a Google product. TalkBack, which it breaks along with every other screen reader, is a Google product. Touch exploration stopping system-wide until you reboot is, by any definition anyone could come up with, broken functionality. I didn't just fall within their scope. I was the textbook example of it.

But the very next paragraph explained that "technical development, custom accessibility service coding, and custom Android framework patches fall outside the scope of our support," so they couldn't review the patches or confirm any of the framework fixes I'd proposed.

I never asked Disability Support to review a patch. I asked them, four times now, to pass a bug report to the people who own the code, and to give me a way to send them the logs. The patches were offered to an engineering team, because that's what you do with a fix: you hand it to the people who can use it. "Custom accessibility service coding" means Backtalk, I assume, which has nothing to do with the bug, as I'd said in the first message, and the second, and the third. The bug happens with TalkBack. Google's TalkBack, running on Google's Android, using Google's touch exploration.

So the more work I did, the further outside their scope the case became. Report a bug with logs, and you're told to clear a cache. Show that the cache had nothing to do with it, and you're told you need to do the troubleshooting first. Do the troubleshooting, find the cause, and write the fix, and you're told that fixes aren't their department. At no point in that loop was there a step where the report went to an engineer, and there was never going to be one. I just kept looking for it.

Then the reply sent me some links. The Android Developers accessibility guide, which is documentation for people making their apps work with screen readers, and is about as relevant to a bug in the input dispatcher as a recipe book. A page of additional accessibility resources. And the Android Accessibility Help Community, where, it said, I could "share your findings, discuss touch exploration behavior, and connect with other accessibility developers." That's a forum. It's other users, answering each other's questions for free, which is exactly what has been happening with this bug for years, and exactly why it's still there.

Nowhere in the email was the issue tracker mentioned. I'd sent them the link in my last message. By the time they wrote this, it had been with an engineer for a day and a half. Whoever reviewed my case either didn't see the link, or saw it and decided that the right thing to tell a blind developer whose bug was already assigned to a Google engineer was to go and post about it on a help forum.

And then it closed with the line that, in the end, says it all: "If you encounter specific broken functionality involving Google's accessibility features or would like to suggest an enhancement to improve accessibility, please feel free to contact us again."

I did. That's what this whole thing was.

## What this tells you

I said in the first post that I didn't blame the agents, and I still don't. The second one was genuinely decent, and I'd guess the final email wasn't really theirs at all: it reads like the internal team's verdict, passed down for someone on the front line to sign. These are people working inside a system, and the system decides what they're allowed to do.

So here's what that system is for, now that I've seen it all the way through.

Google Disability Support is a help desk for using Google's accessibility features. If you can't find a setting, it'll find it for you. If TalkBack is talking too fast, it'll tell you where to slow it down. It may well be good at that, and that's worth something. But it isn't a route for reporting accessibility bugs to Google, and it can't be, because nothing in it leads to anyone who can fix one. When a blind person tells it that something is broken, it can tell them how to turn things off and on again, and once that's run out, it can tell them it's out of scope. Then it sends them to a forum.

What it doesn't do is the one thing a support channel exists for: take a report of something broken and put it in front of the people who can fix it.

## Now imagine it wasn't me

It's tempting to read all this as a story with a happy ending. The bug got to an engineer in the end, and I got there by going round Disability Support entirely. So the lesson is to file on the issue tracker, isn't it? Next time, skip the help desk.

No. Fucking no.

Think about who actually hits this bug. Not a developer with adb and a laptop and a working knowledge of the input pipeline. A blind person, maybe new to Android, maybe older, maybe someone whose phone is the main way they read their post and talk to their family and get a taxi, who one day can't find anything on the screen by touch. They don't know what TouchExplorer is. They've never heard of the input dispatcher. They don't know there's a fake finger stuck on their screen, and they have no way to find out, because every tool that would tell them needs a computer, a cable, developer options and a command line.

So they do what Google tells them to do. They contact Disability Support, and they say: "My phone stopped reading things when I touch the screen. Swiping still works. Restarting fixes it but it keeps happening."

That's a perfectly good bug report. It's the symptom, the workaround and the fact that it recurs. It's exactly what any user anywhere should be expected to give, and it's honestly more than most people manage.

And we now know exactly what happens to it. They get nine questions. They get told to turn TalkBack off and on again, with a warning to have a sighted person nearby. They get told to update Android Accessibility Suite and clear its cache, and they do it, carefully, step by step, because they trust that the people telling them know what they're doing. They reboot, and it works, and they report back that it works, and the case is closed as resolved. A

week later it happens again, and they start over. Or they don't, because they've learned that it's pointless, and they just reboot their phone a couple of times a day for the rest of its life, assuming that's what using Android blind is like.

And if they don't take the reboot as an answer, if they push, the best they can hope for is the email I got. Out of scope. Go and read the developer documentation. Ask on the forum.

Should that person be expected to find the Android issue tracker? Make an account, find the right component out of hundreds, and write a report that an engineer will take seriously, with a device fingerprint and a reproduction? With logs they have no way to capture? Of course not. That's not a reasonable thing to ask of a user, and asking it of a disabled user, through the support channel that exists specifically for disabled users, is obscene. Nobody expects a sighted person whose phone screen stops responding to go and file a bug on a developer tracker. They expect Google, or Samsung, or the shop they bought it from, to deal with it. Turning that work over to the people who are least able to do it, and calling it out of scope when they come asking, is the whole problem in one sentence.

That's what makes it worse than my experience. I had the skills to route round the system, and it still took four days, a blog post, a root cause analysis and two patches. Everyone else just has the system. And the system's answer to "touch exploration is broken" is, in the end, that touch exploration being broken isn't its problem.

So no, the answer isn't "file on the tracker". The answer is that Disability Support should be the thing that files on the tracker. It's Google. It has the same access to the same internal bug systems as every other part of the company. When a disabled user reports that a Google accessibility feature is broken, somebody in that team should write it up, with whatever the user could tell them, file it with the team that owns the feature, and tell the user it's been filed. That's the job. That's what "escalation" is supposed to mean. If the team doesn't have the training, the tools, or the authority to do it, that isn't a reason to tell users it's out of scope. It's a reason to fix the team.

Google points disabled people at Disability Support. It's the channel on the help pages, the human, specialist, accessible route, the place to go when something doesn't work for you. And when one of us arrives with something that doesn't work for any of us, it has no idea what to do with it. Not because the people are bad at their jobs, but because the job, as Google has defined it, stops just short of where it would be any use.

## If you've hit this bug

If you're a blind Android user and touch exploration has died on you while swipes and taps kept working, that's this bug, and it's not your fault, and it's not your screen reader. Rebooting clears it for now.

This one, at least, is filed: TouchExplorer leaves injected pointers down; InputDispatcher then rejects all touch exploration events until reboot. If you have a Google account and want to, sign in and give it a +1. If you can add something, such as a device that isn't a Samsung, a Pixel especially, an Android version, or what you were doing

when it happened, add a comment. You shouldn't have to do that either. But it's the one place this report has actually reached a person who can fix it, and the more people say it hits them, the higher it goes.

And if you've hit any other accessibility bug and you've been told it's out of scope, you weren't wrong to report it, and you weren't asking for the wrong thing. You were asking the right people to do their job.

Quite simply, fuck you, Google. Fix the bug, and then fix the route, so the next blind person who finds one doesn't have to do your engineers' job to be heard.

Subscribe via RSS / via Email.

Powered by mataroa.blog.