


Android Public Tracker37017033

← ↻ ☆ Only one instance of FileObserver is notified


+122Hotlists (11)Mark as Duplicate🔔⋮

Comments (18)DependenciesDuplicates (0)Blocking (0)Resources (2)

AssignedBugP3+

 STATUS UPDATE No update yet.

Edit



DESCRIPTION to...@gmail.com created issue [#1](#)


1. Create two instances of FileObserver to observe the same file for ALL_EVENTS.
2. Call startWatching() on both instances.
3. Modify or delete observed file.

Current result:
Only one (the most recently registered) instance of FileObserver is notified.

Expected result:
All registered and watching instances of FileObserver should be notified.


Android SDK version 20.
Tested on Samsung Galaxy S II (Android 4.1.2) and Samsung Galaxy Note 10.1 Edition 2014 (Android 4.4.2).

I'm attaching slightly modified FileObserverTest.java file. I added second instance of observer. The test fails with AssertionError in line 98: assertTrue(mObserver1.totalEvents > 3).

 deleted


0 B🔗

✓ Mentioned issues (1)✓ Links (1)

 Mentioned issues (1)


P1

Multiple Multi process DataStores not dispatching update ["b/279997241"](#)

 Links (1)


" ...back. We assure you that we are doing our best to address all issues reported. For now, we will be closing the issue as won't fix obsolete. If this issue currently still exists, we request that you log ;

COMMENTS




to...@gmail.com

<to...@gmail.com>

 deleted


0 B🔗



en...@google.com

<en...@google.com>

Assigned to en...@google.com.




na...@google.com

<na...@google.com>

[#2](#)

Uh oh. This is going to be quite painful to fix. The problem is that we use a singleton inotify fd for all our watches. Calling inotify_add_watch with the same path will modify the existing watch design.

enh: mind if i take this on ? i'm getting a bit tired of all the icu & apache changes :)




en...@google.com

<en...@google.com>

[#3](#)

Reassigned to na...@google.com.

be my guest. the only part i was interested in was "can we lose the JNI to libcore?"




ge...@gmail.com

<ge...@gmail.com>

[#4](#)

[Comment deleted]



fi...@gmail.com

<fi...@gmail.com>

[#5](#)

Any update on this? I just got bitten by it, very difficult to find bug, since the first FileObserver fails silently...

Reading through the source, it doesn't seem difficult to fix. The problem is that inotify allows for only one watch per path and inotify-instance (which is consistent with the FileObserver-class) instead of one per FileObserver instance as suggested by the API.

So a simple fix would be to

- make the s_observerThread variable, local to the FileObserver class (instead of static)
- move the two init-lines from the static block to the class' constructor
- introduce an additional native "stopObserving" method called from FileObserver's finalize-block to kill the observing thread
- probably need to make the native code also non-static (method_onEvent variable, etc)

Introduces a bit of overhead since each FileObserver gets its own thread but that's consistent with the inotify setup. Alternatively you could run one thread with non-blocking reading of multip

Could you comment when this will be fixed? Would try to send a patch myself but I don't have AOSP set up here....

fi...@gmail.com <fi...@gmail.com> [#6](#)

Add-on: For anybody bitten by this bug who cannot wait for a fix: I wrote a drop-in replacement for FileObserver that works around the problem by maintaining a global list of FileObservers. If least - haven't tested this super-extensively).



na...@google.com <na...@google.com>

Reassigned to ps...@google.com.

ps...@google.com <ps...@google.com>

Reassigned to to...@google.com.

fi...@gmail.com <fi...@gmail.com> [#7](#)

any updates? ETA?

to...@google.com <to...@google.com> [#8](#)

No ETA because this is P3 and we have no current concrete plan to fix it.

to...@google.com <to...@google.com>

Reassigned to li...@google.com.

as...@gmail.com <as...@gmail.com> [#9](#)

I thought it's my side of code that's not working. Apparently it's a 5 years (and still counting) old bug. Any workaround?

ar...@gmail.com <ar...@gmail.com> [#10](#)

If you can afford API 26 — there is `java.nio.file.WatchService`.

sa...@google.com <sa...@google.com> [#11](#)

Status: Won't Fix (Obsolete)

Thank you for your feedback. We assure you that we are doing our best to address all issues reported. For now, we will be closing the issue as won't fix obsolete. If this issue currently still ex

bd...@google.com <bd...@google.com> [#12](#)

Status: Assigned (reopened)

Reopening issues that had been marked with hotlistid:4240 to prevent bulk closure.

yb...@google.com <yb...@google.com> [#13](#)

Comment has been deleted.

Message last modified on Aug 4, 2023 09:28AM

yb...@google.com <yb...@google.com> [#14](#)

FYI this is affecting Jetpack Datastore.

We'll workaround it w/ an abstraction but it is still risky if the application has other observers that we don't know about.

b/279997241



vv...@gmail.com <vv...@gmail.com> [#15](#)

Will this bug be fixed in Android 15? If it's already an API, will this behavior be documented?



vv...@gmail.com <vv...@gmail.com> [#16](#)

Comment has been deleted.

Message last modified on Aug 4, 2023 11:59AM



vv...@gmail.com <vv...@gmail.com> [#17](#)

This bug has been idle for eight years due to P3, has FileObserver been de facto deprecated, and should we be using WatchService? It was added six years ago



yb...@google.com <yb...@google.com> [#18](#)

Unfortunately, DataStore is still on API 14 so it will be a while before we can rely on WatchService but we can consider using it when available.