Android Public Tracker > Framework   36916605 ▾

← C ☆ system_server consumes excessive CPU processing touch events   +1 412   Hotlists (4)   Mark as Duplicate   🔔 ⋮

Comments (66)   Dependencies   Duplicates (0)   Blocking (0)   Resources (12)

Fixed   Bug   P3   + Add Hotlist   [AOSP] assigned   [AOSP] Released   [AOSP] Version-2.1

👥 **STATUS UPDATE** No update yet. Edit

📄 **DESCRIPTION** rb...@gmail.com created issue #1   Apr 19, 2010 05:27PM ⋮

Summary: system_server uses abnormally high CPU processing touch (motion) events, particularly when the touch point is being moved.

Evidence:
The test is extremely easy to set up. Create a skeleton app with a single hello world activity. You can optionally override onTouchEvent and return true or false or sleep or wait or do whatever the devs have suggested to help with performance. None of it matters. What you will see if you run shell->top is a cpu usage of system_server hovering in a normal range of 5-10% on 1st gen devices or lower on newer devices but while touching, up to 36%.

My personal testing has shown usage up to:
36% CPU usage on my G1/1.6
30% CPU usage on my Droid/2.1
26% CPU usage on my Nexus One/2.1
30% CPU usage on a Milestone/2.0.1

Besides not being power efficient to consume one quarter to one third of the entire CPU on simply filtering and passing touch coordinates to the application layer, it is very bad for games, especially OpenGL ones. 50% loss of frame rate is common and has been observed in many of the current games on the market. The cause is clearly this.

Files affected:
frameworks/base/services/jni/com_android_server_KeyInputQueue.cpp
frameworks/base/services/java/com/android/server/InputDevice.java
frameworks/base/services/java/com/android/server/KeyInputQueue.java

Problems observed in code:
1) Failure to follow posted Android optimizations (including excessive use of virtuals inside of nested loops, etc)
2) Excessive amounts of processing which could be done in native code

This has been documented in great detail on android-developers. Many developers have mistaken the event flooding as the root cause but that has been resolved and the CPU usage problem remains. It is much more evident on devices using chips like the MSM7200 and is truly crippling for game control systems on that platform.

As far as game development is concerned, this is not a "would like to have fixed" issue but more of a "if this is not fixed, we can't deploy on anything but really high-end devices" sort of an issue. I've spoken with several iphone developers who can't port their games because of this problem. I've also spoken to several Android game developers who have abandoned games because of this same issue.

Because of all this, I believe this should be a very high priority issue.

Thank you.

✓ Links (12)   Hide all

"http://groups.google.com/group/android-developers/browse_thread/thread/39eea4d7f6e6dfca#..."   ko...@ #7
"http://www.reddit.com/r/Android/comments/bsz8r/please_star_at_this_bug_if_you..."   ko...@ #7
"http://stackoverflow.com/questions/792185/why-are-touch-events-destroying-..."   ko...@ #7
"http://www.google.com/codesearch/p?hl=de#uX1GffpyOZk/services/java/com/android/server/KeyInputQueue.java&q=KeyI..."   ba...@ #8
"http://www.youtube.com/watch?v=ECI4MpVIpXw"   d....@ #14

See all related links

COMMENTS   All comments ▾   ↓ Oldest first

⊙ **ba...@gmail.com** <ba...@gmail.com> #2   Apr 19, 2010 07:06PM ⋮

Reporter

Type

Priority

Severity

Status

Access

Assignee

Verifier

Collaborators

CC

AOSP ID

ReportedBy

Found In

Targeted To

Verified In

In Prod

I can reproduce this problem as well. This is a classic me too comment, just trying
to add that little bit of weight i can add to the topic. Along with the multi-touch
issue (which is said to be a hardware problem) this is one of the biggest problems
for Android game developers.

ba...@gmail.com <ba...@gmail.com> #3   Apr 19, 2010 07:07PM ⋮

[Comment deleted]

ed...@gmail.com <ed...@gmail.com> #4   Apr 19, 2010 10:48PM ⋮

I can confirm this as well. Framerate on my game drops from ~30fps idling/moving via
orientation sensor to ~10-15fps after a user touches the screen to make the
character jump.

ke...@gmail.com <ke...@gmail.com> #5   Apr 19, 2010 10:55PM ⋮

priority level should be bumped up IMHO...it really needs to be fixed in a 2.5
release (assuming the rumor of this version is true)...

an...@gmail.com <an...@gmail.com> #6   Apr 20, 2010 12:21AM ⋮

I too can reproduce this issue. I am one of the developers that has abandoned game
development because of this issue among a few, this being the highest offender for
me. Being in the industry and knowing others from various channels, I know I am not
alone in those leaving the android platform due this and a few other plaguing issues.
If this issue is not bumped up and fixed as soon as possible, I fear android has
little chance of competing against iPhone and will even go as far as saying that I
would bet that the new microsoft xna based mobile devices coming out will surpass
android in market share. Given that games are almost 50% of the iPhone market, I
would hope that google goes out of it's way to put more resources on issues like
these to get android on par with iPhone/MS mobile devices so that us game developers
can take android to that next level.

ko...@gmail.com <ko...@gmail.com> #7   Apr 20, 2010 01:37AM ⋮

Developers: try adding a sleep() timeout as per Dianne Hackborn's suggestion.[1]
Sheep from Reddit[2], you could also find the answer on StackOverflow[3].

[1]
http://groups.google.com/group/android-developers/browse_thread/thread/39eea4d7f6e6dfca#msg_c1d2c2c4950ef14d
[2]
http://www.reddit.com/r/Android/comments/bsz8r/please_star_at_this_bug_if_you_want_faster_games/
[3]
http://stackoverflow.com/questions/792185/why-are-touch-events-destroying-my-android-framerate

ba...@gmail.com <ba...@gmail.com> #8   Apr 20, 2010 01:49AM ⋮

Have a look at this

http://www.google.com/codesearch/p?hl=de#uX1GffpyOZk/services/java/com/android/server/KeyInputQueue.java&q=KeyInputQueue&sa=N&cd=1&ct=rc&l=492

The event handling seems to be done in a seperate thread. Now, i'm far from being an
Android platform expert but if this is true it means that sleeping in your UI thread
will only prevent getting flooded with events in your UI but the original problem of
CPU overutilization is still there

rb...@gmail.com <rb...@gmail.com> #9   Apr 20, 2010 02:30AM ⋮

kostmo.

That workaround has proven ineffective. Please read the bug description. Many of us
have tested every suggested workaround including sleep of various length, thread
reprioritization, and a thread wait/notify/yield technique. What you will find is if
you do nothing but sleep(32) and return false, you will see the overutilization and
your real-time app will suffer.

Let's be careful not to confuse this with the "event flood" issue that was fixed in
2.0, where more than 60 motion events per second were fired, requiring a management
workaround on the UI thread as you linked to. This is NOT that issue but probably is
much of what devs were noticing as slowdown during that time.

ko...@gmail.com <ko...@gmail.com> #10   Apr 20, 2010 03:51AM ⋮

My bad, I did not read the description carefully enough.

fr...@gmail.com <fr...@gmail.com> #11

Apr 20, 2010 04:34AM

I can also reproduce this problem. Nothing more to add than what's already been said.
THe priority of this bug should be set to 'high'.

---

es...@googlemail.com <es...@googlemail.com> #12

Apr 20, 2010 06:38AM

Because of this we don't publish our rotary dialer app. The framerate while touching and dialing is not
acceptable. All well known workarounds (sleeping, pausing the ui-thread, ...) won't make the app work smooth.
Raising the app's -thread/-process-priority only heats the device up. Compared to iPhone rotary dialer apps our
app isn't fun. So this is nothing I would pay for and 'Dialuah' is on hold. An app should be able to inform the
touch- (and trackball) input-layer about its priority similar to the SensorManager
(SensorManager.SENSOR_DELAY_...) The priority of this bug should be set to 'high'.

---

vm...@gmail.com <vm...@gmail.com> #13

Apr 20, 2010 03:20PM

Vote for better android event handling, I have noticed this by toggling on the cpu
watcher while touching and moving on the screen.

---

d....@gmail.com <d....@gmail.com> #14

Apr 23, 2010 05:15AM

I took the liberty and made a video which shows the discussed behaviour:

http://www.youtube.com/watch?v=ECl4MpVIpXw

Handset is HTC Magic/myTouch3g, Android 1.6

If someone with a 2nd gen device (Droid/N1) wants to recreate this bug, the app is
called Show CPU Usage and can be acquired for free in the market. (see here for QR
code:
http://www.androidpit.com/en/android/market/apps/app/yuku.coba.cpuusage/Show-CPU-Usage).

[Deleted User] <[Deleted User]> #15

May 1, 2010 01:02AM

Interestingly, this bug does not seem to occur on the HTC Incredible. No noticeable CPU usage changes using the
above application.

---

rb...@gmail.com <rb...@gmail.com> #16

May 1, 2010 01:16AM

snorpapotamous - This bug is primarily about the first gen devices based on the MSM7200
chip. That chip + an OpenGL game + this problem = bad user experience. Please test on
a G1, Eris, Cliq, Behold, MyTouch, Hero or other Qualcomm MSM7200-based device to see
the issue.

Thanks

---

am...@gmail.com <am...@gmail.com> #17

May 1, 2010 01:34AM

Actually this bug exists on the Motorola Droid running the ESE81 2.1 update also:

http://www.youtube.com/watch?v=gi2Y-5CVjnM

---

la...@gmail.com <la...@gmail.com> #18

May 1, 2010 01:55AM

HTC Incredible uses a new sensor.
http://www.ellismarkov.com/android-devices-and-touch-capability

---

ja...@gmail.com <ja...@gmail.com> #19

May 1, 2010 11:25AM

Just as an extra data point, I have an Acer Liquid with Android 1.6. My idle CPU
hovers around 3% (1-5%), actively moving finger on touchscreen makes it jump around
and average 6% (4-9%).

---

w....@gmail.com <w....@gmail.com> #20

May 1, 2010 12:48PM

I just tested this bug on my UK Milestone 2.01 and I can confirm the same behaviour. I ran a couple of small
experiments to better understand the nature of this bug:

1) When repeatedly tapping the screen to simulate quick typing, cpu usage peaks to around 7%. (From a base
of 1-3)
2) When moving a single touch point on the screen the cpu seems to work in the range 15-22% based on
which motion is selected.
3) When two fingers are added cpu usage spikes up to the high twenties, however I found it was easier to push

this up by adding a third touch point.

Obviously all this depends on the nature of the movements, how rapid they are, and how many points are touching the screen. But it would explain some small cpu spikes I have noticed in every day use of the swype application.

---

sr...@gmail.com <sr...@gmail.com> #21　　　　　　　　　　　　　　　　　May 5, 2010 02:19PM　⋮

I'd like to also add my name to the list of devs trying to deal with this issue. I'm really affected by this issue in my real-time game because I'm utilizing canvas for all the graphics. My game doesn't lend itself to openGl so I'm already taking a performance hit. Everything I can do to squeeze out a much fps is important to me. My frame rates on my G1 are 30+ fps without touching the screen and drops to around 20 fps while dragging on the screen. I'm building to 1.6 currently

---

ja...@gmail.com <ja...@gmail.com> #22　　　　　　　　　　　　　　　　　May 11, 2010 02:30PM　⋮

On Acer Liquid E 2.1 ROM (0.014f Beta), 768 Mhz Qualcomm 8250 Snapdragon CPU, idle 3% CPU, gesturing bumps it up to around 20%. On Original 1.6 ROM it didn't use as much CPU (10% or less when gesturing)

---

co...@gmail.com <co...@gmail.com> #23　　　　　　　　　　　　　　　　　May 31, 2010 04:48AM　⋮

The priority should be high in my opinion. With this one fix, the whole user experience would be much much better.

The very easy way to reproduce and monitor this, is to watch top while you're holding your finger on the screen.

I can also confirm the bug on my hero in several roms (1.5 and 2.1).

---

bo...@gmail.com <bo...@gmail.com> #24　　　　　　　　　　　　　　　　　May 31, 2010 07:15AM　⋮

Is it fixed in Froyo?

---

rb...@gmail.com <rb...@gmail.com> #25　　　　　　　　　　　　　　　　　May 31, 2010 07:26AM　⋮

The usage needs to be tested on an MSM7200-based device to be of any use. If someone can run froyo on that chip, the test results would be very helpful. My guess is that the usage will be lower. How much lower is key.

---

ar...@gmail.com <ar...@gmail.com> #26　　　　　　　　　　　　　　　　　Jun 12, 2010 01:53PM　⋮

I'm running the Samsung Moment with Android 2.1 which has an 800MHz ARM11 processor. Faster than the MSM7200 devices but slower than a Droid/Milestone.

I ran adb shell -> top. CPU usage was around 1% to 4% with the screen off. With the screen on and my finger pressed on the home screen it shot up to 6% going up to 12%.

Not bad, but here's what surprised me the most: Holding my finger on the home screen and sliding SLOWLY left and right, system_server shot up to 40% to 50% and android.process.acore added another 30% to 45% causing an average load of 70% from just sliding the screen back and forth.

I had a huge WTF face on when I saw the screen refresh showing those numbers. Come on Google, you're better than this.

Also, there's a game called AirHockeyGL on the Market. I was so embarrassed when I downloaded the game and showed it off to a friend who had an iPhone. The puck lagged and shot across the screen making me lose track of it and constantly lose. What's worse is the phone doesn't have multitouch so I can't play against others.

---

ap...@gmail.com <ap...@gmail.com> #27　　　　　　　　　　　　　　　　　Jun 18, 2010 05:06AM　⋮

I have the HTC Droid Incredible and I am not affected by this Issue. I ran adb shell: top and system_server CPU usage was 1% with no touch and peaked at 2% when I moved my finger around.

I've never toyed with the Android source code, but my intuition tells me this might be hardware/driver related.

This is a big ouch for Google and I'm really curious why this isn't assigned to anyone, I think this may be the most important issue affecting Android.

---

br...@google.com <br...@google.com> #28　　　　　　　　　　　　　　　　　Jun 18, 2010 05:20AM　⋮

The input system code has all been rewritten lately, as you'll notice in the latest comments from Jeff Brown here:

https://review.source.android.com/#change,15073

---

rb...@gmail.com <rb...@gmail.com> #29　　　　　　　　　　　　　　　　　Jun 18, 2010 05:39AM　⋮

What would be ideal is if the issues could be resolved or at least improved and then backported to an update that all first gen devices could receive, even those that can not get up to 2.2. Perhaps it's asking for a lot but this specific problem has a lot to do with user experience issues gaming on Android.

ap...@gmail.com <ap...@gmail.com> #30   Jun 18, 2010 08:30AM   ⋮

My friend received a magazine today from T-Mobile "100% You" (haha, right) in it, T-Mobile commits to having an OTA update for the MyTouch with 2.2, so I think this could be the solution we need. (Lets Hope)

---

lu...@gmail.com <lu...@gmail.com> #31   Jul 6, 2010 11:18AM   ⋮

Well, tested today with CyanogenMod nightly 2.2 and this bug is still present.
So, I think that 2.2 will not solve this issue for these devices lol

---

ap...@gmail.com <ap...@gmail.com> #32   Jul 8, 2010 04:32AM   ⋮

The "Target" is set to "Gingerbread", so realistically we can't assume this problem will be fixed for at least the next release cycle, since Froyo is coming out soon. This is a huge problem for Google (and all phone manufacturers).

@bradfitzgoog: thank you for your updates, it has given us a glimmer of hope. Can you communicate to us the status of this issue? I believe this may be the highest priority bug affecting the Android clients right now.

---

bo...@gmail.com <bo...@gmail.com> #33   Jul 23, 2010 12:26PM   ⋮

The guys at VillainROM.co.uk have a fix for this in their latest 2.2 build for the HTC Hero. There's still a slight drop in frame rate but nowhere near as bad. Neocore is over 25fps during constant touch input and Raging Thunder 2 is now playable during touch input.

---

bh...@gmail.com <bh...@gmail.com> #34   Aug 31, 2010 05:04PM   ⋮

In Froyo on x86, we are seeing similar issue. The system_server is taking around 30% to 95% of CPU. Due to this, the response of the menu and other basic application also getting affected.

---

ra...@gmail.com <ra...@gmail.com> #35   Aug 31, 2010 10:58PM   ⋮

Does anybody know any trust work around to this issue before google provide a solution?

---

ap...@gmail.com <ap...@gmail.com> #36   Sep 23, 2010 02:24AM   ⋮

[Comment deleted]

---

go...@gmail.com <go...@gmail.com> #37   Oct 3, 2010 08:27PM   ⋮

Are there any new on this? Nook device with Android 1.5 raises CPU usage to ~60% while touching..

---

lu...@gmail.com <lu...@gmail.com> #38   Oct 4, 2010 12:02AM   ⋮

@gorbuz yes. Read comments. Still present on Android 2.2+

---

sr...@gmail.com <sr...@gmail.com> #39   Oct 6, 2010 09:59AM   ⋮

Glad to see this issue has been assign to someone to fix. Jeff, good luck to you!!!

---

do...@gmail.com <do...@gmail.com> #40   Oct 6, 2010 01:19PM   ⋮

I second that! Thank you Jeff!

---

ap...@gmail.com <ap...@gmail.com> #41   Oct 6, 2010 01:22PM   ⋮

Third that!

---

lu...@gmail.com <lu...@gmail.com> #42   Oct 7, 2010 04:52AM   ⋮

ETA?

ehhe just kidding. Good news are good.

---

ru...@gmail.com <ru...@gmail.com> #43   Oct 22, 2010 05:42AM   ⋮

I have a Samsung Galaxy Spica (i5700) and can also reproduce this issue. I am doing a drag&drop game and this hurts performance really badly :(

Good Luck Jeff

os...@googlemail.com <os...@googlemail.com> #44   Nov 22, 2010 04:37AM ⋮

One Day into android programming, and already such a showstopper (on my LG Optimus One).

ap...@gmail.com <ap...@gmail.com> #45   Nov 22, 2010 05:52AM ⋮

Yes, I have a lot of bad feedback on an otherwise 5-star game. This is really disappointing.

ne...@gmail.com <ne...@gmail.com> #46   Dec 2, 2010 08:08PM ⋮

LG Optimus One - 100% CPU. It is not possible to play games where you have to click on the screen :(

du...@gmail.com <du...@gmail.com> #47   Dec 3, 2010 12:24AM ⋮

I'd like to add myself to the long list of people this is affecting. I am about to release a game, but risk getting negative reviews from users of affected phones who will have a bad experience.

It's great that someone has been assigned to this, but has anyone heard anything from Google or the manufacturers regarding what developers (particularly games developers) are supposed to do in the meantime?

Is it something that can be fixed in software for devices that are already out there or will affected devices never be able to run my game?

Can I put a disclaimer in my game's description and in the game itself if I detect this issue, to try and limit the number of comments I will inevitably get?

If users complain to me about this or demand a refund what can I tell them? Will there eventually be an update released for the devices that are currently affected?

I understand that issues of this kind will sometimes slip through QA and make it into devices, and that the fixes do take time. But in these situations there is usually some form of communication from the platform providers about when a fix will likely be available, what form it will take, and what we should do in the meantime (apart from just abandon the platform altogether).

If anyone has information to share I'd be very grateful.

je...@google.com <je...@google.com> #48   Dec 4, 2010 12:58PM ⋮

Let's just say it is much much better. More details later. :)

ba...@gmail.com <ba...@gmail.com> #49   Dec 4, 2010 01:08PM ⋮

i assume there's no way of backporting it. i don't think any of the problematic devices will get an update to 2.3.

sr...@gmail.com <sr...@gmail.com> #50   Dec 4, 2010 02:03PM ⋮

@Jeff

Please, please, please fill us in. I'm really hoping to hear whatever progress large or small you may have made on this issue. For me, even if I have to target 2.2 and up with my apps, thats a small price to pay for getting this sorted out.

ap...@gmail.com <ap...@gmail.com> #51   Dec 4, 2010 02:24PM ⋮

Awesome, Thank you ;)

te...@gmail.com <te...@gmail.com> #52   Dec 6, 2010 08:49PM ⋮

LG Optimus One same problem-)

am...@gmail.com <am...@gmail.com> #53   Dec 6, 2010 08:58PM ⋮

LG Optimus One. Please solve this problem!

je...@gtempaccount.com <je...@gtempaccount.com> #54   Dec 7, 2010 06:49AM ⋮

Marked as fixed.

The Gingerbread SDK was released today!

One of the major new features in Gingerbread is a complete rewrite of the input dispatch pipeline. The CPU usage and latency for processing touches and keys has been significantly improved. These benefits apply to all applications running on Gingerbread. In other words, your applications will process input faster without requiring any extra effort on your part when running on a Gingerbread based device.

In Gingerbread, we also added new NDK APIs for receiving input from native code. Together with the native activity lifecycle, sensors, audio and graphics APIs, it is now possible to write native applications that contain very little Dalvik code of your own (or even none at all).

Here are some other new features that you might find interesting.

MotionEvents now carry the size of the touch contact area in pixels. Look for the new touchMajor, touchMinor, toolMajor, toolMinor and orientation properties. You can easily test this using the Pointer Location tool by pressing a conductive object of known size like a penny up to the screen. On a properly calibrated device, the circle or ellipse that represents the touch area will closely circumscribe the penny.

The pressure calibration is also somewhat better. Applications can distinguish typical small featherlight brushes from full finger touches. However, since the device can't actually measure pressure directly we use a simple approximation based on the signal strength. The approximation is inaccurate but it turns out to be good enough for some purposes. The Touch Paint API Demo has been upgraded to demonstrate this feature.

If you have any additional questions about the new APIs and input dispatch in general, please feel free to contact me on the android-platform mailing list.

Cheers!

---

r1...@gmail.com <r1...@gmail.com> #55   Dec 7, 2010 07:00AM   ⋮

Thank you very much. You give happiness to people.

---

ap...@gmail.com <ap...@gmail.com> #56   Dec 7, 2010 07:09AM   ⋮

Hi, this is awesome!

Can this be made available to pre-gingerbread devices? Say those running Donut or Froyo? Considering this graph: http://developer.android.com/resources/dashboard/platform-versions.html they make up a considerable number of devices (most of which will never see gingerbread, or not at least for 6-12 months).

---

rb...@gmail.com <rb...@gmail.com> #57   Dec 7, 2010 07:12AM   ⋮

I think I can speak for all Android game developers when I say.. THANK YOU!!!

---

lu...@gmail.com <lu...@gmail.com> #58   Dec 7, 2010 07:33AM   ⋮

That is all very beautiful words but let's see if somehow i get that on my HTC Magic lol I am at risk that never get this fix!

---

ve...@gmail.com <ve...@gmail.com> #59   Dec 29, 2010 06:28PM   ⋮

IN LG optimus ONE
100% CPU usage on a stok Froyo 2.2 and 2.2.1

---

se...@gmail.com <se...@gmail.com> #60   May 14, 2011 02:02AM   ⋮

So, what should do ppl with Froyo? As I understand there no chance it will be backported.
Any workaround avail?

---

am...@gmail.com <am...@gmail.com> #61   May 14, 2011 05:17AM   ⋮

See LG P500 patch, on xda, but it's not good fix. This patch adds sleep for
input events
13.05.2011 19:03 ������������ <android@googlecode.com> ������:

excessive CPU processing touch events

be backported.

---

la...@gmail.com <la...@gmail.com> #62   Oct 3, 2011 11:53PM   ⋮

Nexus S: all kind of roms have the same issue.

---

ma...@gmail.com <ma...@gmail.com> #63   Jun 4, 2012 10:49AM   ⋮

I need,help. My email address is maria.graciela.soto26@gmail.com. I downloaded by mistake internet Explorer in my android tablet and now some information appear in the top right corner. I don't know to restored an deleted this. Could you help me.? Thanks

---

da...@gmail.com <da...@gmail.com> #64   Jan 29, 2015 06:42AM   ⋮

A

---

ia...@gmail.com <ia...@gmail.com> #65   Jun 8, 2015 07:57AM   ⋮

I'm having issues with system server on android wear (Sony Smartwatch 3)...

```
 PID PR CPU% S #THR   VSS    RSS PCY UID     Name
 421 0 52% S  75 677588K 117276K  fg system  system_server
```

And this just starts happening now and then, lets just say that the watch gets warm...

And logcat continiously spews:

```
W/BroadcastQueue( 421): Exception when sending broadcast to
ComponentInfo{com.google.android.gms/com.google.android.gms.common.stats.GmsCoreStatsServiceLauncher}
W/BroadcastQueue( 421): android.os.TransactionTooLargeException
W/BroadcastQueue( 421):     at android.os.BinderProxy.transactNative(Native Method)
W/BroadcastQueue( 421):     at android.os.BinderProxy.transact(Binder.java:496)
W/BroadcastQueue( 421):     at android.app.ApplicationThreadProxy.scheduleReceiver(ApplicationThreadNative.java:861)
W/BroadcastQueue( 421):     at com.android.server.am.BroadcastQueue.processCurBroadcastLocked(BroadcastQueue.java:245)
W/BroadcastQueue( 421):     at com.android.server.am.BroadcastQueue.processNextBroadcast(BroadcastQueue.java:898)
W/BroadcastQueue( 421):     at com.android.server.am.ActivityManagerService.finishReceiver(ActivityManagerService.java:16430)
W/BroadcastQueue( 421):     at android.app.ActivityManagerNative.onTransact(ActivityManagerNative.java:469)
W/BroadcastQueue( 421):     at com.android.server.am.ActivityManagerService.onTransact(ActivityManagerService.java:2208)
W/BroadcastQueue( 421):     at android.os.Binder.execTransact(Binder.java:446)
E/libprocessgroup( 421): failed to kill 1 processes for processgroup 30361
I/ActivityManager( 421): Start proc 30382:com.google.android.gms/u0a10 for broadcast com.google.android.gms/.common.stats.GmsCoreStatsServiceLauncher
W/ResourcesManager(30382): Asset path '/system/framework/com.android.location.provider.jar' does not exist or contains no resources.
I/MultiDex(30382): VM with version 2.1.0 has multidex support
I/MultiDex(30382): install
I/MultiDex(30382): VM has multidex support, MultiDex support library is disabled.
I/ProviderInstaller(30382): Insert disabled by gate 'gms:security:enable_conscrypt_in_gms_application'
V/GmsCoreStatsServiceLauncher(30382): Received broadcast intent Intent { act=com.google.android.gms.INITIALIZE flg=0x10 pkg=com.google.android.gms
cmp=com.google.android.gms/.common.stats.GmsCoreStatsServiceLauncher }
I/ActivityManager( 421): Killing 30382:com.google.android.gms/u0a10 (adj 9): 21931k from cached
E/libprocessgroup( 421): failed to kill 1 processes for processgroup 30382
E/JavaBinder( 421): !!! FAILED BINDER TRANSACTION !!!
```

in...@gmail.com <in...@gmail.com> #66   Nov 16, 2015 02:14AM ⋮

priority level should be bumped up IMHO...it really needs to be fixed in a 2.5
release (assuming the rumor of this version is true)...
http://www.wdfshare.com