Please add a public API to receive ANR notifications

+1 Hotlists (2) Mark as Duplicate

Comments (3) Dependencies Duplicates (0) Blocking (0) Resources (8)

Obsolete Feature Request P3 + Add Hotlist [AOSP] assigned

STATUS UPDATE No update yet. Edit

DESCRIPTION ka...@bugsnag.com created issue #1 Jan 22, 2021 09:27PM

The current design of Android's ANR detection mechanism makes it extremely difficult for users to capture ANR events without disrupting Google's ANR detection.

https://android.googlesource.com/platform/art/+/master/runtime/signal_catcher.cc#154

The runtime library uses sigwait() to capture SIGQUIT signals. sigwait() only works when the signal is blocked, so the runtime library must also block SIGQUIT before starting the app.

The standard mechanism that everyone uses for signal catching is sigaction(), which won't trigger if the signal is blocked. Upon discovering that SIGQUIT is blocked, most users will simply unblock it, not realising that this breaks the runtime library's ANR detection. Even worse, they'll probably follow the best-practice of forwarding the signal to the next signal handler in the chain, which in this case is the default handler that quits the app, turning an ANR into an immediate app shutdown.

Capturing SIGQUIT safely in a way that doesn't break Google's ANR handler is actually very complicated and difficult to get right. You must:

- Find and store the process ID and the thread ID of Google's "Signal Catcher" thread (to make a syscall later)
- Unblock SIGQUIT (temporarily breaking Google SIGQUIT code, so that your SIGQUIT handler will run)
- Install a SIGQUIT handler using sigaction()

In your SIGQUIT handler:

- Re-block SIGQUIT so that the Google code will get triggered when SIGQUIT becomes pending.
- Pass control to another thread so that the OS signal mechanism has time during the context switch to update the new signal blocking state.
- In the other (non-signal-handler) thread, raise a signal directly on Google's "Signal Catcher" thread using a syscall (raise() and signal() won't work).

An example here:

- Code to get the runtime ANR thread ID
- Code to forward SIGQUIT to the runtime handler

Almost nobody knows of the existence of sigwait() or its implications, so writing end-user code that doesn't clobber Google's ANR detection is incredibly difficult to get right. I suspect that most software that attempts to capture ANR events is doing it wrong without realising it, and preventing ANR events from reaching Google Play.

The runtime library does have a callback mechanism for SIGQUIT, but there's no public API to access it.

- Call to notify callbacks
- Method Runtime::GetRuntimeCallbacks()
- Class RuntimeSigQuitCallback

It would be great to have a public mechanism for registering to receive ANR events. The code is mostly there already, so it would simply be a matter of creating a public API for it.

COMMENTS All comments Oldest first

bn...@google.com <bn...@google.com> #2 Jan 22, 2021 10:01PM

Assigned to bn...@google.com.

Thanks for the feedback. We have passed this issue onto our product and engineering teams and we will update this issue with more information as it becomes available.

bn...@google.com <bn...@google.com> #3 Feb 16, 2021 06:02PM

Status: Won't Fix (Obsolete)

Thank you for your feedback. We will be closing this Feature request as "Won't Fix - Obsolete". Please note, we have informed the respective internal engineering team to further look into this feature. Please keep a watch on the future android releases.

Reporter	ka...@bugsnag.com
Type	Feature Request
Priority	P3
Severity	S3
Status	Won't fix (Obsolete)
Access	Default access View
Assignee	bn...@google.com
Verifier	--
Collaborators	
CC	al...@google.com bn...@google.com ka...@bugsnag.com
AOSP ID	--
ReportedBy	Developer
Found In	--
Targeted To	--
Verified In	--
In Prod	