

← ↺ ☆ Investigate retry mechanism in androidx.sqlite to reduce SQLITE_BUSY

+1

Hotlists (1)


Mark as Duplicate






Comments (5)DependenciesDuplicates (0)Blocking (0)Resources (10)

AssignedFeature RequestP3+ Add Hotlist

 STATUS UPDATE No update yet.

Edit

 DESCRIPTION da...@google.com created issue [#1](#)

This is useful for WorkManager and its multi-process support. See [b/196281488](#)

Right now the framework already has some built-in retry. There is a default 2.5 seconds set via [↔️sqlite3_busy_timeout](#) and the other is when iterating over a statement, there is a [↔️0.05 sec](#)

We should investigate adding retry mechanisms to the transactions APIs, specifically beginTransaction().

Also note that with an unbundled SQLite we can configure the timeout (or provide an API for configuring it) as we like.

✓ Mentioned issues (2)✓ Links (8)

 Mentioned issues (2)

- "...is useful for WorkManager and its multi-process support. See [b/196281488](#)"
- "... I found out that execPerConnectionSQL is a bit problematic with this pragma, see [b/197128690](#)."

↔️ Links (8)

"Right now the framework already has some built-in retry. There is a default 2.5 seconds set via [↔️sqlite3_busy_timeout](#) and the other is when iterating over a statement, there is a 0.05 seconds retry.

"...mework already has some built-in retry. There is a default 2.5 seconds set via sqlite3_busy_timeout and the other is when iterating over a statement, there is a [↔️0.05 seconds retry](#)."

"...Lite added the [↔️busy_timeout PRAGMA](#) in SQLite 3.7.15 released in 2012-12-12 and available in API 21 and above. An application could use this pragma to increase the framework's 2.5 seconds "

"... added the busy_timeout PRAGMA in SQLite 3.7.15 released in 2012-12-12 and [↔️available in API 21](#) and above. An application could use this pragma to increase the framework's 2.5 seconds time

"[https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase#execPerConnectionSQL\(java.lan...](#)"

See all related links

COMMENTS

 da...@google.com <da...@google.com> [#2](#)

SQLite added the [↔️busy_timeout PRAGMA](#) in SQLite 3.7.15 released in 2012-12-12 and [↔️available in API 21](#) and above. An application could use this pragma to increase the framework's SQLITE_BUSY.

 da...@google.com <da...@google.com> [#3](#)

The busy_timeout is a per-connection setting, so one would need to use it via the [↔️execPerConnectionSQL](#) API that is only available in API 30. Otherwise using it in [↔️onConfigure](#) will only also used by transactions).

Sadly I found out that `execPerConnectionSQL` is a bit problematic with this pragma, see [b/197128690](#).

 ap...@google.com <ap...@google.com> [#4](#)

Project: platform/frameworks/support
Branch: androidx-main

commit 3c871590837878c0148ab1ec8567396f0e415fc3
Author: Daniel Santiago Rivera <[danyssantiago@google.com](#)>
Date: Wed Mar 02 09:14:55 2022

Enable 'data loss on recovery' for WorkManager

Enabling this option will let Room and androidx.sqlite specifically delete the database during its last database open retry. This option is available in androidx.sqlite 2.3.0-alpha hence the ch

Bug: 196412170
Test: Existing
Change-Id: I0b98eadb2c12350b217761b5def706009162c4eb

M work/work-runtime/build.gradle
M work/work-runtime/src/main/java/androidx/work/impl/WorkDatabase.kt

[https://android-review.googlesource.com/2005012](#)



ap...@google.com <ap...@google.com> [#5](#)

Project: platform/frameworks/support
Branch: androidx-main

commit 6af02899721a2747cbe585ad40a0b5f742d2f176
Author: Elif Bilgin <elifbilgin@google.com>
Date: Mon Mar 07 14:31:05 2022

Changing `project` dependency on sqlite in work/work-runtime to `projectOrArtifact`. Follow up to the change on aosp/2005012.

Test: Existing
Bug: 196412170
Change-Id: Ia3d3ec49d23a7c69273aa1d4f5bdbeaee823eeb4

M work/work-runtime/build.gradle

<https://android-review.googlesource.com/2012218>