

覚え書き

2011年3月11日金曜日

SQLiteDatabase is closed automatically under some conditions

To say the result of investigation first, "A database (`android.database.sqlite.SQLiteDatabase`) used to create a cursor (`android.database.Cursor`) is automatically closed if some simple conditions are satisfied."

The flow which triggers the auto close is as follows.

1. Actual data fetch is triggered when either `Cursor.getCount()` method or `Cursor.onMove()` method is called.
2. It triggers `SQLiteQuery.fillWindow(CursorWindow, int, int)` method to be called.
3. `SQLiteQuery.fillWindow()` calls `SQLiteQuery.releaseReferences()` as its last step.
4. The implementation of `SQLiteQuery.releaseReference()` is in its indirect super class, `SQLiteClosable`. (`SQLiteQuery` extends `SQLiteProgram`, and `SQLiteProgram` extends `SQLiteClosable`.)
5. `SQLiteClosable.releaseReference()` decrements the reference count.
6. If the reference count becomes 0, `SQLiteClosable` calls an abstract method, `onAllReferencesReleased()`.
7. The super class of `SQLiteQuery`, `SQLiteProgram`, has the implementation of `onAllReferencesReleased()`.
8. `SQLiteProgram.onAllReferencesReleased()` calls `releaseReference()` method on an `SQLiteDatabase` instance it holds.
9. `SQLiteDatabase` also extends `SQLiteClosable`. As a result, `SQLiteDatabase.onAllReferencesReleased()` is called.
10. `SQLiteDatabase.onAllReferencesReleased()` calls a native method, `dbclose()`.
11. `dbclose()` closes the underlying database by calling `sqlite3_close()`.

Therefore, the coding like below:

- i. Create a cursor.
- ii. Create a cursor adapter using the created cursor.
- iii. Pass the cursor adapter to a `ListView` using `ListView.setAdapter()` method.

causes the database (`SQLiteDatabase`) to be closed automatically because the implementation of `ListView.setAdapter()` calls `getCount()` method of the given cursor and so the steps from (1) to (11) are performed. On the contrary, if the step (1) is not started after a cursor is created, the auto close won't occur and the database remains open.

If you encounter one of the following error messages, there is a possibility that the error is caused by the database auto close. (These are error messages that are contained in the current or past implementations of `SQLiteClosable.acquireReference()` method.)

```
"attempt to re-open an already-closed object: " + getObjInfo()
"attempt to acquire a reference on an already-closed " + getObjInfo()
"attempt to acquire a reference on an already-closed SQLiteClosable obj."
"attempt to acquire a reference on a close SQLiteClosable"
```

If you want to prevent the database from being automatically closed, one possible solution would be to call `SQLiteClosable.acquireReference()` to prevent the reference count of the database from getting down to 0. In such a case, the database needs to be closed later manually, of course.

ブログ アーカイブ

▶ 2012 (15)

▼ 2011 (30)

▶ 12月 (3)

▶ 11月 (1)

▶ 9月 (2)

▶ 8月 (2)

▶ 7月 (3)

▶ 6月 (3)

▶ 5月 (1)

▶ 4月 (2)

▼ 3月 (4)

Usage of `dx --dex`
(`dx --dex` の使い方)

SQLiteDatabase
is closed
automatically
under some ...

ある条件下で SQLiteDatabase は自動的にクローズされる

調査結果を先に言うと、「カーソル（`android.database.Cursor`）作成時に使用したデータベース（`android.database.sqlite.SQLiteDatabase`）は、ある要件を満たすと、自動的にクローズされる。」

自動クローズが発生するときの処理の流れは次のようになっている。

1. カーソルの `Cursor.getCount()` メソッドもしくは `onMove()` メソッドが呼ばれると、実際のデータ取得処理が走る。
2. これにより `SQLiteQuery.fillWindow(CursorWindow, int, int)` メソッドが呼ばれることになる。
3. `SQLiteQuery.fillWindow()` は、最後に `SQLiteQuery.releaseReference()` を呼ぶ。
4. `SQLiteQuery.releaseReference()` の実体は、親（`SQLiteProgram`）の親クラスの `SQLiteClosable` にある。
5. `SQLiteClosable.releaseReference()` は、参照カウントを減らす。
6. 参照カウントがゼロになると、`SQLiteClosable` は抽象メソッド `onAllReferencesReleased()` を呼ぶ。
7. `SQLiteQuery` の親クラス `SQLiteProgram` が `onAllReferencesReleased()` を実装している。
8. `SQLiteProgram.onAllReferencesReleased()` は、保持している `SQLiteDatabase` のインスタンスの `releaseReference()` メソッドを呼ぶ。
9. `SQLiteDatabase` も `SQLiteClosable` を継承しているので、結果、`SQLiteDatabase.onAllReferencesReleased()` が呼ばれる。
10. `SQLiteDatabase.onAllReferencesReleased()` は、ネイティブメソッド `dbclose()` を呼ぶ。
11. `dbclose()` は、`sqlite3_close()` を呼んでデータベースを閉じる。

そういうわけで、


- i. カーソルを作成する。
- ii. 作成したカーソルをもとにカーソルアダプタを作成する。
- iii. カーソルアダプタをリストビューの `setAdapter()` メソッドを用いてリストビューに渡す。

というコーディングをすると、リストビューの `setAdapter()` メソッドの実装がカーソルの `getCount()` メソッドを呼ぶので、結果上記 (1)～(11) の処理が走り、データベース (`SQLiteDatabase`) は自動的にクローズされることになる。逆に言うと、カーソルを作成したものの、上記 (1) が発生しないと、データベースの自動クローズ処理は走らないので、放っておくとリークしてしまう。

もしも次のいずれかのエラーメッセージに遭遇したなら、データベースの自動クローズが走ってしまったことが原因の可能性がある。（これらは過去もしくは現在の `SQLiteClosable.acquireReference()` の実装に含まれていたエラーメッセージである。）

```
"attempt to re-open an already-closed object: " + getObjInfo()
"attempt to acquire a reference on an already-closed " + getObjInfo()
"attempt to acquire a reference on an already-closed SQLiteClosable obj."
"attempt to acquire a reference on a close SQLiteClosable"
```

自動でクローズさせたくない場合は、一つの解決方法として、データベースの参照カウントがゼロにならないように `SQLiteClosable.acquireReference()` を呼ぶという手がある。もちろん後で自分でデータベースをクローズする必要がある。

投稿者 [だる](#) 時刻: 5:57 

+1 Google でおすすめする

0 件のコメント:

[コメントを投稿](#)

Android エミュレータが圏外状態になる問題（解決？）

Android `System.exit(int)`: 驚愕の実装

▶ 2月 (3)

▶ 1月 (6)

▶ 2010 (17)

自己紹介

[だる](#)

President @ Neo Visionaries Inc.

[詳細プロフィールを表示](#)

コメントを入力...

コメントの記入者: Google アカウント ▼

公開

プレビュー

[次の投稿](#)

[ホーム](#)

[前の投稿](#)

登録: [コメントの投稿 \(Atom\)](#)

Awesome Inc. テンプレート. Powered by [Blogger](#).