CheckJNI always flags ByteBuffer with binary data in AndroidJUnit4

STATUS UPDATE No update yet. Edit

DESCRIPTION kn...@gmail.com created issue #1

Class with native method:

```
import java.nio.ByteBuffer

class SomeClass {
  companion object {
    init { System.loadLibrary("thelib") }
  }

  external fun funWith(randomBytes: ByteBuffer)
}
```

"Unit" Test:

```
@RunWith(AndroidJUnit4::class)
class ExampleInstrumentedTest {
  @Test
  fun testFunWith() {
    val some = SomeClass()
    val rand = SecureRandom()
    val bytes = ByteArray(32)
    rand.nextBytes(bytes)
    val byteBuffer = ByteBuffer.allocateDirect(32)
    byteBuffer.put(bytes)
    byteBuffer.flip()
    some.funWith(byteBuffer)
  }
}
```

I'll spare you the native side. I can't get past this error:

```
JNI DETECTED ERROR IN APPLICATION: input is not valid Modified UTF-8: illegal start byte 0x98
     string: 'F��c5q|V�# ��C �Ӵ��]�1�$� ��'Ӈ � pH� �$�0�'
     input: '0x46 <0x98> 0x98 0x63 0x35 0x71 0x7c 0x56 0xb1 0x23 0x19 0x89 0xf4 0x43 0x0e 0xf9 0xd3 0xb4 0xd4 0xf3 0x5d 0xf5 0x31 0xb4 0x24 0xcb 0x19 0xf8 0x
     in call to GetMethodID
     from SomeClass.funWith(java.nio.ByteBuffer)
```

```
Build: AI-222.4459.24.2221.10121639, 202305120703,

AI-222.4459.24.2221.10121639, JRE 17.0.6+0-17.0.6b802.4-9586694x64 JetBrains s.r.o., OS Mac OS X(x86_64) v13.4, screens 3840.0x2160.0, 3840.0x2400.0; Retina

AS: Flamingo | 2022.2.1 Patch 2
Kotlin plugin: 222-1.8.20-release-AS3739.54
Android Gradle Plugin: 8.0.2
Gradle: 8.0
Gradle JDK: JetBrains Runtime version 17.0.6
NDK: from module: 25.2.9519653, from local.properties: (not specified), latest from SDK: 22.1.7171670
CMake: from local.properties: (not specified), latest from SDK: 3.22.1-g37088a8, from PATH: 3.24.4
```

COMMENTS

gh...@google.com <gh...@google.com> #2

> in call to GetMethodID

Can you paste the relevant parts of the native code too? I'm unsure whether you call `GetMethodID`, or if Android Runtime is doing this under the hood.

Also, you mention that this happens in `AndroidJUnit4`. Do you see a similar issue in production code too? Or only in `AndroidJUnit4` tests specifically?

ba...@gmail.com <ba...@gmail.com> #3

What's this about

kn...@gmail.com <kn...@gmail.com> #4

My fault! ...thanks for the GetMethodID clue. I still learning to parse the signal from everything that dumped to the console; I didn't notice that in the stack; and all the googling I did about the error pointed to the JNI being made. It was working fine, but I was doing something foolish on the native side with that call to GetMethodID, and I didn't expect CheckJNI to operate there.

Sorry for the clutter and waste of time!

On Fri, Jun 23, 2023 at 11:04 PM <buganizer-system@google.com> wrote:

- Show quoted text -

ad...@google.com <ad...@google.com>

Status: Won't Fix (Intended Behavior)

gh...@google.com <gh...@google.com> #5

No worries, thanks for confirming!