Renderscript doesn't work on x86 devices

Fixed Bug P3 + Add Hotlist [AOSP] FutureRelease

STATUS UPDATE No update yet. Edit

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

Using Renderscript from the support library in our app to blur images we receive the following exception on x86 devices like the Motorola Razr i (XT 890) and the Genymotion emulator with differ

```
06-12 15:02:55.415  14724-14724/my.android.app E/RenderScript_jni : No GC methods
06-12 15:02:55.415  14724-14724/my.android.app D/dalvikvm : Trying to load lib /data/data/net.lovoo.android/lib/libRSSupport.so 0x420878c0
06-12 15:02:55.425  14724-14724/my.android.app D/dalvikvm : Added shared lib /data/data/net.lovoo.android/lib/libRSSupport.so 0x420878c0
06-12 15:02:55.425  14724-14724/my.android.app D/dalvikvm : No JNI_OnLoad found in /data/data/net.lovoo.android/lib/libRSSupport.so 0x420878c0, skipping init
06-12 15:02:55.425  14724-14724/my.android.app D/dalvikvm : Trying to load lib /data/data/net.lovoo.android/lib/librsjni.so 0x420878c0
06-12 15:02:55.435  14724-14724/my.android.app E/RenderScript_jni : Error loading RS jni library: java.lang.UnsatisfiedLinkError: Cannot load library: reloc_library[1306]:   141 cannot locate '__s
06-12 15:02:55.435  14724-14724/my.android.app D/AndroidRuntime : Shutting down VM
06-12 15:02:55.435  14724-14724/my.android.app W/dalvikvm : threadid=1: thread exiting with uncaught exception (group=0x4184c7d0)
06-12 15:02:55.445  14724-14724/my.android.app E/RSRuntimeException : android.support.v8.renderscript.RSRuntimeException: Error loading RS jni library: java.lang.UnsatisfiedLinkError: Can
        at android.support.v8.renderscript.RenderScript.create(RenderScript.java:945)
        at android.support.v8.renderscript.RenderScript.create(RenderScript.java:982)
        at android.support.v8.renderscript.RenderScript.create(RenderScript.java:968)
        at my.android.app.transformations.BlurImageTransformation.<init>(BlurImageTransformation.java:41)
```

We're building the app using the current 19.1.0 build tools and the 0.11.1 Android gradle plugin. Renderscript config is as following:

```
renderscriptTargetApi 19
renderscriptSupportMode true
```

On armv7 devices everything works well.

We found the following question on Stackoverflow which seems to address the same issue, but contains no real solution:

http://stackoverflow.com/questions/22040821/error-loading-rs-jni-library-unsatisfiedlinkerror-couldnt-load-rssupport-fin

In the comments regarding this question Tim Murray stated he is (was?) working on this issue, but there has been no news since then. We consider this is a major blocker as excluding x86 devic

✓ Links (4)

“http://my.android.app”
“http://stackoverflow.com/questions/22040821/error-loading-rs-jni-library-unsatisfiedlinkerror-c… ”
“http://eefilters.o.info”
“ …issing "dot" function for x86 should be fixed with https://android-review.googlesource.com/#/c/179305/1 (which I just merged into AOSP). Unfortunately, I think that the only way to work around th

COMMENTS

en...@google.com <en...@google.com>

Assigned to ti...@google.com.

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

Marked as fixed.

This has been fixed internally and will be released in the next version of the build tools.

[Deleted User] <[Deleted User]> #3

[Comment deleted]

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

The update has already been released. I can confirm that everything works on x86 devices now. :)

[Deleted User] <[Deleted User]> #5

Android Build Tools version 20 is out - but there is no 19.1.1, yet. If I understand this correctly, that's what we'd need to make RenderScript run on Intel platform.

Is there an estimate when this fix would be released?

ub...@gmail.com <ub...@gmail.com> #6

Hm, why wouldn't you want to use 20.0.0? It incorporates the Renderscript libs build for x86 devices. I doubt they will release a 19.1.1 after having released 20.0.0.

[Deleted User] <[Deleted User]> #7

I'm trying to bend my mind over how this versioning works, so thanks for any help.

The main reason would be to avoid changing the targeted API level. We're aiming KitKat and don't want to rock the boat with L, yet.

May I ask what values you have in the project.properties (i.e. can we target android-19 but pull in renderscript.target=20 and/or sdk.buildtools=20).

target=android-19
android.library=false

renderscript.target=19
renderscript.support.mode=true

sdk.buildtools=20

What is the combo you use?

ub...@gmail.com <ub...@gmail.com> #8

Using build tools versions higher than your target SDK version shouldn't be a problem - AFAIK build tools versioning is only loosely coupled to Android API versions (maybe one of the Andro
renderscript.target=19 (imho this should always match the Android API level you are targeting) and target=android-19 should be just fine for your case. In fact our configuration looks very sir

ub...@gmail.com <ub...@gmail.com> #9

Ahem, yeah, now that's embarrassing... Maybe someone in our team accidentally commented the lines applying the blur effects and checked that into our repository causing everything to wo
build tools version... I just found that lines in our code and readded the blurring code which in turn led to the aforementioned crashes on x86 devices. So we are where we where at the beginn

Any news from a Google dev about when this will be fixed?

ti...@google.com <ti...@google.com> #10

sorry all, this was supposed to be fixed in build tools v20, but the change didn't make it to the right place somehow. I've attached libs that should have the fix--put them in sdk/build-tools/and

if these work or don't work, please post in the bug and I'll take another look.

I'll follow up with the SDK team to see if we can push a v20.0.1 sometime soon.

deleted
0 B

deleted
0 B

[Deleted User] <[Deleted User]> #11

Thank you, Tim

We needed to place the files in the ../20.0.0/... branch instead of android-4.4W but other than that we're now okay to proceed.

yx...@gmail.com <yx...@gmail.com> #12

[Comment deleted]

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

This bug still exists when using Android Studio 0.8.11. After updating my build.gradle with "renderscriptTargetApi 19" and "renderscriptSupportMode true" I still get the same error reported al

sr...@google.com <sr...@google.com> #14

#13. Did you update your buildtools to 20 or higher? The Target API is not the part that is an issue here.

ub...@gmail.com <ub...@gmail.com> #15

There is only build tools 20 out there - 20.0.1 which should fix the issue according to #10 is unfortunately still not released. So the problem persists - regardless of the API version.

[Deleted User] <[Deleted User]> #16

In build tools 21.1.0 the fix isn't still released

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

For us 21.1.0 fixed the problem.

[Deleted User] <[Deleted User]> #18

The size of file libRSSupport.so remained the same since version 20 of build-tools

[Deleted User] <[Deleted User]> #19

Not working on 21.1.1

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

Not working on 21.1.2

[Deleted User] <[Deleted User]> #21

still happening on build tools 21.1.2
I've been running some tests and I'll put logs below.

We really would appreciate some attention on the renderscript as we're having a lot of users out there crashing on it, we bought a x86 device for testing, but all we could do was add the code

```
if (Build.CPU_ABI.contains("x86")) // skip renderscript stuff
```

Before each test I uninstalled the app from the device and did a full clean-rebuild to avoid any cache related issues.

If I execute the code a second time I usually get the added message:
Invalid RS info file /data/data/com.baseapp.eyeem/cache/com.android.renderscript.cache/eefilters.o.info! (No such file or directory)

Test device: Asus Zenfone - ASUS_T00J - Android 4.4.2 - build ASUS_T00J_WW_user_2.20.40.13_20141105_1055 (latest)
gradle: compile SDK 21, target SDK 21, minSDK 15, renderscriptTargetApi 18, renderscriptSupport true
CreateFxThread is the thread we're using to instantiate all the RS related stuff.

using buildTools 20 with original binaries:

```
================================================================================================================
01-08 09:02:56.265  15934-16042/com.baseapp.eyeem.p0 E/RenderScript﹕ Skipping override driver 'libPVRRS.so' and loading default CPU driver 'libRSDriver.so'.
01-08 09:02:56.285  15934-16042/com.baseapp.eyeem.p0 D/RenderScript﹕ Load RS driver 'libRSDriver.so' successfully.
01-08 10:24:28.905  23716-23822/com.baseapp.eyeem.p0 A/libc﹕ Fatal signal 11 (SIGSEGV) at 0x00000010 (code=1), thread 23822 (CreateFxThread)
01-08 10:24:28.955     179-179/? I/DEBUG﹕ *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
01-08 10:24:28.955     179-179/? I/DEBUG﹕ Build fingerprint: 'asus/WW_a501cg/ASUS_T00J:4.4.2/KVT49L/WW_user_2.20.40.13_20141105_1055:user/release-keys'
01-08 10:24:28.955     179-179/? I/DEBUG﹕ Revision: '0'
01-08 10:24:28.955     179-179/? I/DEBUG﹕ pid: 23716, tid: 23822, name: CreateFxThread  >>> com.baseapp.eyeem.p0 <<<
01-08 10:24:28.955     179-179/? I/DEBUG﹕ signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 00000010
01-08 10:24:28.995     179-179/? I/DEBUG﹕ eax 658732b8  ebx 4288dd3c  ecx 0010be19  edx 00000000
01-08 10:24:28.995     179-179/? I/DEBUG﹕ esi 668077a0  edi 670beab4
01-08 10:24:28.995     179-179/? I/DEBUG﹕ xcs 00000073  xds 0000007b  xes 0000007b  xfs 00000000  xss 0000007b
01-08 10:24:28.995     179-179/? I/DEBUG﹕ eip 42468b14  ebp 658732c4  esp 65873278  flags 00210216
01-08 10:24:28.995     179-179/? E/Corkscrew﹕ unrecognized dwarf lower part encoding: 0x54
01-08 10:24:28.995     179-179/? I/DEBUG﹕ backtrace:
01-08 10:24:28.995     179-179/? I/DEBUG﹕ #00  pc 00a7db14  /system/lib/libLLVM.so
01-08 10:24:28.995     179-179/? I/DEBUG﹕ #01  pc 0010117e  /system/lib/libbcc.so
01-08 10:24:28.995     179-179/? I/DEBUG﹕ stack:
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 65873238  00000000
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 6587323c  00000000
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 65873240  6587353c  [stack:23822]
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 65873244  65873234  [stack:23822]
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 65873248  418a57b9  /system/lib/libbcc.so
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 6587324c  419658e0  /system/lib/libbcc.so
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 65873250  66296e14  [anon:libc_malloc]
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 65873254  67867eb4  [anon:libc_malloc]
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 65873258  658732c4  [stack:23822]
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 6587325c  418a58ee  /system/lib/libbcc.so
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 65873260  66296e14  [anon:libc_malloc]
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 65873264  658732c4  [stack:23822]
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 65873268  418a4269  /system/lib/libbcc.so
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 6587326c  40468850  /system/lib/libstlport.so
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 65873270  419658e0  /system/lib/libbcc.so
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 65873274  42468b0e  /system/lib/libLLVM.so
01-08 10:24:28.995     179-179/? I/DEBUG﹕ #00  65873278  419658e0  /system/lib/libbcc.so
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 6587327c  418f717f  /system/lib/libbcc.so
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 65873280  658732b8  [stack:23822]
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 65873284  00000000
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 65873288  00000013
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 6587328c  00000000
01-08 10:24:28.995     179-179/? I/DEBUG﹕ 65873290  0000000e
```

```
01-08 10:24:28.995  179-179/? I/DEBUG ﹕ 65873294  42895c54
01-08 10:24:28.995  179-179/? I/DEBUG ﹕ 65873298  00000013
01-08 10:24:28.995  179-179/? I/DEBUG ﹕ 6587329c  65175e48  [anon:libc_malloc]
01-08 10:24:28.995  179-179/? I/DEBUG ﹕ 658732a0  66739e40  [anon:libc_malloc]
01-08 10:24:28.995  179-179/? I/DEBUG ﹕ 658732a4  65175e5b  [anon:libc_malloc]
01-08 10:24:28.995  179-179/? I/DEBUG ﹕ 658732a8  6799d890  [anon:libc_malloc]
01-08 10:24:28.995  179-179/? I/DEBUG ﹕ 658732ac  6799d8a4  [anon:libc_malloc]
01-08 10:24:28.995  179-179/? I/DEBUG ﹕ 658732b0  0088dd3c
01-08 10:24:28.995  179-179/? I/DEBUG ﹕ 658732b4  419282b8  /system/lib/libbcc.so
```

using buildTools 20 with modified binaries supplied on this thread:

```
================================================================================================================
01-08 10:33:48.405  24216-24378/com.baseapp.eyeem.p0 E/RenderScript ﹕ Skipping override driver 'libPVRRS.so' and loading default CPU driver 'libRSDriver.so'.
01-08 10:33:48.405  24216-24378/com.baseapp.eyeem.p0 D/RenderScript ﹕ Load RS driver 'libRSDriver.so' successfully.
01-08 10:33:49.015  24216-24378/com.baseapp.eyeem.p0 A/libc ﹕ Fatal signal 11 (SIGSEGV) at 0x00000010 (code=1), thread 24378 (CreateFxThread)
01-08 10:33:49.075  179-179/? I/DEBUG ﹕ *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
01-08 10:33:49.075  179-179/? I/DEBUG ﹕ Build fingerprint: 'asus/WW_a501cg/ASUS_T00J:4.4.2/KVT49L/WW_user_2.20.40.13_20141105_1055:user/release-keys'
01-08 10:33:49.075  179-179/? I/DEBUG ﹕ Revision: '0'
01-08 10:33:49.075  179-179/? I/DEBUG ﹕ pid: 24216, tid: 24378, name: CreateFxThread  >>> com.baseapp.eyeem.p0 <<<
01-08 10:33:49.075  179-179/? I/DEBUG ﹕ signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 00000010
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ eax 665b12b8  ebx 4288dd3c  ecx 002533ff  edx 00000000
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ esi 68fb9fa0  edi 67862bf4
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ xcs 00000073  xds 0000007b  xes 0000007b  xfs 00000000  xss 0000007b
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ eip 42468b14  ebp 665b12c4  esp 665b1278  flags 00210216
01-08 10:33:49.115  179-179/? E/Corkscrew ﹕ unrecognized dwarf lower part encoding: 0x54
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ backtrace:
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ #00  pc 00a7db14  /system/lib/libLLVM.so
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ #01  pc 0010117e  /system/lib/libbcc.so
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ stack:
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b1238  00000000
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b123c  00000000
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b1240  665b153c  [stack:24378]
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b1244  665b1234  [stack:24378]
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b1248  418a57b9  /system/lib/libbcc.so
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b124c  419658e0  /system/lib/libbcc.so
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b1250  66fad234  [anon:libc_malloc]
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b1254  678629a4  [anon:libc_malloc]
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b1258  665b12c4  [stack:24378]
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b125c  418a58ee  /system/lib/libbcc.so
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b1260  66fad234  [anon:libc_malloc]
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b1264  665b12c4  [stack:24378]
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b1268  418a4269  /system/lib/libbcc.so
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b126c  40468850  /system/lib/libstlport.so
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b1270  419658e0  /system/lib/libbcc.so
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b1274  42468b0e  /system/lib/libLLVM.so
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ #00  665b1278  419658e0  /system/lib/libbcc.so
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b127c  418f717f  /system/lib/libbcc.so
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b1280  665b12b8  [stack:24378]
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b1284  00000000
01-08 10:33:49.115  179-179/? I/DEBUG ﹕ 665b1288  00000013
01-08 10:33:49.125  179-179/? I/DEBUG ﹕ 665b128c  00000000
01-08 10:33:49.125  179-179/? I/DEBUG ﹕ 665b1290  0000000e
01-08 10:33:49.125  179-179/? I/DEBUG ﹕ 665b1294  42895c54
01-08 10:33:49.125  179-179/? I/DEBUG ﹕ 665b1298  00000013
01-08 10:33:49.125  179-179/? I/DEBUG ﹕ 665b129c  66ff2068  [anon:libc_malloc]
01-08 10:33:49.125  179-179/? I/DEBUG ﹕ 665b12a0  68a18720  [anon:libc_malloc]
01-08 10:33:49.125  179-179/? I/DEBUG ﹕ 665b12a4  66ff207b  [anon:libc_malloc]
01-08 10:33:49.125  179-179/? I/DEBUG ﹕ 665b12a8  669e1d90  [anon:libc_malloc]
01-08 10:33:49.125  179-179/? I/DEBUG ﹕ 665b12ac  669e1da4  [anon:libc_malloc]
01-08 10:33:49.125  179-179/? I/DEBUG ﹕ 665b12b0  0088dd3c
01-08 10:33:49.125  179-179/? I/DEBUG ﹕ 665b12b4  419282b8  /system/lib/libbcc.so
```

using buildTools 21.1.2:

```
================================================================================================================
01-08 10:46:04.255  25189-25396/com.baseapp.eyeem.p0 E/RenderScript ﹕ Skipping override driver 'libPVRRS.so' and loading default CPU driver 'libRSDriver.so'.
01-08 10:46:04.275  25189-25396/com.baseapp.eyeem.p0 D/RenderScript ﹕ Load RS driver 'libRSDriver.so' successfully.
```

and on this point nothing happens (no crash), but it also doesn't process any script, I open on AndroidDeviceMonitor and found our `CreateFxThread` stuck in a native method inside the cons
the stack:

```
at android.support.v8.renderscript.ScriptC.<init>(ScriptC.java:62)
at android.support.v8.renderscript.ScriptCThunker.<init>(ScriptCThunker.java:39)
at android.renderscript.ScriptC.<init>(ScriptC.java:59)
at android.renderscript.ScriptC.internalCreate(ScriptC.java:111)
at android.renderscript.RenderScript.nScriptCCreate(RenderScript.java:702)
at android.renderscript.RenderScript.rsnScriptCCreate(Native Method)
```

---

in...@gmail.com <in...@gmail.com> #22

Still not working. Any update on this please?

---

cs...@gmail.com <cs...@gmail.com> #23

Still not working. Any update on this please?

ul...@gmail.com <ul...@gmail.com> #24

I had a similar problem with the Asus Fonepad 7 K00E tablet.
The problem:
The tablet is able to run both x86 and ARM code. For ARM it uses the houdini library to do the magic. They try to be smart and count the native dynamic libraries for each architecture. If the n
for armeabi so it goes boom.

The workaround:
Add some dummy empty libraries in the jniLibs/armeabi folder. In this way the device will start using the x86 libraries. I attached the make file to create those empty libs and also the libs con

deleted
0 B

deleted
0 B

deleted
0 B

ag...@google.com <ag...@google.com> #25

Please complain to Intel.

bo...@gmail.com <bo...@gmail.com> #26

#25 Is that a joke?

sr...@google.com <sr...@google.com> #27

re: #24 above. If the issue with Houdini is that it sees more libraries for arm, the real issue is that you aren't shipping the appropriate full set of compatibility libraries for your app (i.e. arm,x86
the raw number of libraries, any NEON optimized code or specialized library you add for ARM will potentially trip this. That is a bug for Intel, as #25 mentions, since we have no visibility/acces

ag...@google.com <ag...@google.com> #28

Re #26:

I wish I was. This is an Intel modification to Android - not from Google. Try your app on a Nexus Player, and you will *not* find this behavior.

sa...@gmail.com <sa...@gmail.com> #29

Getting the same issue with the 23.0.0 buildtools:

Caused by: android.support.v8.renderscript.RSRuntimeException: Error loading RS jni library: java.lang.UnsatisfiedLinkError: JNI_ERR returned from JNI_OnLoad in "/data/app/com.mbfg.pop
android.support.v8.renderscript.RenderScript.internalCreate(RenderScript.java:1347)
android.support.v8.renderscript.RenderScript.create(RenderScript.java:1504)
android.support.v8.renderscript.RenderScript.create(RenderScript.java:1454)
android.support.v8.renderscript.RenderScript.create(RenderScript.java:1430)
android.support.v8.renderscript.RenderScript.create(RenderScript.java:1417)

ka...@gmail.com <ka...@gmail.com> #30

This hack works for me (Asus Zenfone 5 T00J, android 4.3 build-tools 22.0.1):

add one fake so file (just make and rename txt file) to build-tools/22.0.1/renderscript/lib/packaged/x86
rebuild project
crashes gone

Unfortunately this doesn't work for 23.0.1 (so I can't use for example 23 support libraries)

[Deleted User] <[Deleted User]> #31

x86 + Renderscript Compat, few remarks:

Most of the time the simple "hello world" for RS compat code will compile and run. If it doesn't then that RS + buildTools version combination is wrong.

With that in mind, the methods and the way you add them is the thing that might break the code on x86. We managed to get our code running after hours of commenting out lines of code and

For us it turned out to be an array of method pointers in our RS code - that didn't work well with x86. Circumventing that with a silly switch fixed it and allowed rest of the code to execute prop

Unfortunately with latest build tools (23.0.1) our code is broken again. There is now the classic™ x86 error appearing:

-- dlopen failed: cannot locate symbol "_Z3dotDv3_fS_"

We had to revert back to 22.0.1 - I dare say there is some x86 regression in 23.0.1 as we haven't changed our rs code.

sr...@google.com <sr...@google.com> #32

The missing "dot" function for x86 should be fixed with https://android-review.googlesource.com/#/c/179305/1 (which I just merged into AOSP). Unfortunately, I think that the only way to wo

le...@gmail.com <le...@gmail.com> #33

The combination of:

compileSdkVersion 23
buildToolsVersion "23.0.2"
renderscriptTargetApi 19
renderscriptSupportModeEnabled true
com.android.support:appcompat-v7:23.1.1

works not too!

```
android.support.v8.renderscript.RSRuntimeException: Error loading libRSSupport library
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at android.support.v8.renderscript.RenderScript.internalCreate(RenderScript.java:1379)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at android.support.v8.renderscript.RenderScript.create(RenderScript.java:1504)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at android.support.v8.renderscript.RenderScript.create(RenderScript.java:1454)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at android.support.v8.renderscript.RenderScript.create(RenderScript.java:1430)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at android.support.v8.renderscript.RenderScript.create(RenderScript.java:1417)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at de.ndr.sfdl.helper.BitmapHelper.blurRenderScript(BitmapHelper.java:21)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at de.ndr.sfdl.activities.MainActivity.showLoading(MainActivity.java:486)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at de.ndr.sfdl.fragments.RegUserFragment.regUserOnBackEnd(RegUserFragment.java:98)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at de.ndr.sfdl.fragments.RegUserFragment.onClick(RegUserFragment.java:74)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at android.view.View.performClick(View.java:4232)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at android.view.View$PerformClick.run(View.java:17298)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at android.os.Handler.handleCallback(Handler.java:615)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at android.os.Handler.dispatchMessage(Handler.java:92)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at android.os.Looper.loop(Looper.java:137)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at android.app.ActivityThread.main(ActivityThread.java:4921)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at java.lang.reflect.Method.invokeNative(Native Method)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at java.lang.reflect.Method.invoke(Method.java:511)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:1027)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:794)
11-13 08:15:26.130 5948-5948/de.ndr.sfdl E/AndroidRuntime:    at dalvik.system.NativeStart.main(Native Method)
```

is there any workaround, so i can use appcompat-v7:23.+ ?

lk...@gmail.com <lk...@gmail.com> #34

[Comment deleted]

lk...@gmail.com <lk...@gmail.com> #35

How to write dot function?

sr...@google.com <sr...@google.com> #36

Here are the functions from the runtime library itself. Just create your own dot function with this:

```
extern float __attribute__((overloadable)) dot(float lhs, float rhs) {
   return lhs * rhs;
}
extern float __attribute__((overloadable)) dot(float2 lhs, float2 rhs) {
   return lhs.x*rhs.x + lhs.y*rhs.y;
}
extern float __attribute__((overloadable)) dot(float3 lhs, float3 rhs) {
   return lhs.x*rhs.x + lhs.y*rhs.y + lhs.z*rhs.z;
}
extern float __attribute__((overloadable)) dot(float4 lhs, float4 rhs) {
   return lhs.x*rhs.x + lhs.y*rhs.y + lhs.z*rhs.z + lhs.w*rhs.w;
}
```

[Deleted User] <[Deleted User]> #37

[Comment deleted]

ai...@gmail.com <ai...@gmail.com> #38

build tools 23.0.2 fixed that crash for me