NE - Run MTBF Test Machine Reboot

WAI Feature Request P3 + Add Hotlist [AOSP] assigned

STATUS UPDATE No update yet. Edit

DESCRIPTION xi...@xiaomi.corp-partner.google.com created issue #1

The M12 development board and stable version have all been reproduced, and the project is nearing package closure. Could you please help expedite the processing. Could you please help chec

Build fingerprint: 'Xiaomi/corot_global/corot:13/TP1A.220624.014/23.05.9-M12-MTBF-PRE-GLO:user/test-keys' Revision: '0' ABI: 'arm64' Timestamp: 2023-05-10 10:34:36.738652770+0800 Proc
tagged_addr_ctrl: 0000000000000001 (PR_TAGGED_ADDR_ENABLE) signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 0x0000000000000010 Cause: null pointer dereference x0 b400007
000000001a58be54 x8 b4000794c1b180 x9 0000000000000002 x10 0000000000000000 x11 490fe01bae08f360 x12 0000000000000004 x13 b4000077b2492180 x14 000000000000000a x
00000077a3c72d60 x22 0000000000000005 x23 a07db5462b2f2a04 x24 0000000000000002 x25 000000001a11b020 x26 000000001a585810 x27 000000001a585cc0 x28 000000001a11b00

backtrace: #00 pc 00000000000bcc84 /system/lib64/libgui.so (std::__1::pair<std::__1::__hash_iterator<std::__1::__hash_node<android::sp android::SurfaceControl, void*>*, bool> std::__1::__has
*std::__1::allocator<android::sp android::SurfaceControl > >::__emplace_unique_key_args<android::sp android::SurfaceControl, android::sp android::SurfaceControl const&>(android::sp a
0000000000117c48 /system/lib64/libgui.so (android::SurfaceComposerClient::Transaction::registerSurfaceControlForCallback(android::sp android::SurfaceControl const&)+120) (BuildId: 0c4
(android::SurfaceComposerClient::Transaction::reparent(android::sp android::SurfaceControl const&, android::sp android::SurfaceControl const&)+152) (BuildId: 0c4899e92a897627fa80014
844680c10f98755dda467c523bfa737f) #04 pc 00000000001e054c /system/framework/arm64/boot-framework.oat (art_jni_trampoline+124) (BuildId: e919b6fd402cc98ebf8684868cff2e859d9
e919b6fd402cc98ebf8684868cff2e859d9925e6) #06 pc 0000000000785218 /system/framework/arm64/boot-framework.oat (android.view.SurfaceControl$Transaction.remove+56) (BuildId: e9
(com.android.server.wm.AppWindowAnimatorHelper.destoryMiuiActivityThumbnailLeash+96) (BuildId: be1204eeb734931495b23cf543e3ca6088b1bbe1) #08 pc 00000000022da744 /system/fr
be1204eeb734931495b23cf543e3ca6088b1bbe1) #09 pc 00000000011d6e74 /system/framework/oat/arm64/services.odex ([DEDUPED]+68) (BuildId: be1204eeb734931495b23cf543e3ca6088
(BuildId: be1204eeb734931495b23cf543e3ca6088b1bbe1) #11 pc 0000000002af77a0 /system/framework/oat/arm64/services.odex (com.android.server.wm.SurfaceAnimator.cancelAnimation
(com.android.server.wm.WindowContainer.cancelAnimation+200) (BuildId: be1204eeb734931495b23cf543e3ca6088b1bbe1) #13 pc 000000000246f1c0 /system/framework/oat/arm64/service
0000000001b330e8 /system/framework/oat/arm64/services.odex (com.android.server.wm.RemoteAnimationController$$ExternalSyntheticLambda3.accept+168) (BuildId: be1204eeb7349314
(com.android.server.wm.ActivityRecord.forAllActivities+76) (BuildId: be1204eeb734931495b23cf543e3ca6088b1bbe1) #16 pc 0000000002ce3654 /system/framework/oat/arm64/services.ode
/system/framework/oat/arm64/services.odex (com.android.server.wm.WindowContainer.forAllActivities+56) (BuildId: be1204eeb734931495b23cf543e3ca6088b1bbe1) #18 pc 00000000022c7
be1204eeb734931495b23cf543e3ca6088b1bbe1) #19 pc 00000000022c516c /system/framework/oat/arm64/services.odex (com.android.server.wm.RemoteAnimationController$FinishedCallb
(android.view.IRemoteAnimationFinishedCallback$Stub.onTransact+336) #21 pc 00000000005f58b0 /system/framework/arm64/boot-framework.oat (android.os.Binder.execTransactInternal+1
(android.os.Binder.execTransact+304) (BuildId: e919b6fd402cc98ebf8684868cff2e859d9925e6) #23 pc 000000000043476c /apex/com.android.art/lib64/libart.so (art_quick_invoke_stub+556)
art::InvokeVirtualOrInterfaceWithVarArgs art::ArtMethod*(art::ScopedObjectAccessAlreadyRunnable const&, _jobject*, art::ArtMethod*, std::__va_list)+828) (BuildId: 28c5aa8a2e8fc5df069f717
std::__va_list)+184) (BuildId: 28c5aa8a2e8fc5df069f717d6e94f7fe) #26 pc 00000000000c47b8 /system/lib64/libandroid_runtime.so (_JNIEnv::CallBooleanMethod(_jobject*, _jmethodID*, ...)+12
android::Parcel const&, android::Parcel*, unsigned int)+156) (BuildId: 844680c10f98755dda467c523bfa737f) #28 pc 0000000000050a4c /system/lib64/libbinder.so (android::BBinder::transact(u
/system/lib64/libbinder.so (android::IPCThreadState::executeCommand(int)+1012) (BuildId: 1d66fcc1bedcffd8a51220cead36f0f4) #30 pc 000000000005b7d0 /system/lib64/libbinder.so (andro
(android::IPCThreadState::joinThreadPool(bool)+68) (BuildId: 1d66fcc1bedcffd8a51220cead36f0f4) #32 pc 000000000008bed8 /system/lib64/libbinder.so (android::PoolThread::threadLoop()+2
263dd89cc6d0dd79143c59615c4821ef) #34 pc 00000000000cd55c /system/lib64/libandroid_runtime.so (android::AndroidRuntime::javaThreadShell(void*)+140) (BuildId: 844680c10f98755dda
3908c7c57fa04c64df24425cf16523cf) #36 pc 000000000008e5f0 /apex/com.android.runtime/lib64/bionic/libc.so (__start_thread+64) (BuildId: 3908c7c57fa04c64df24425cf16523cf)

deleted
0 B

✓ Links (3)

Links (3)

"...e version have all been reproduced, and the project is nearing package closure. Could you please help expedite the processing. Could you please help check if your company has encountered this is
"about:invalid#zCSafez"
"For steps to capture a bug report, please refer: https://developer.android.com/studio/debug/bug-report#bugreportdevice"

COMMENTS

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

Assigned to vi...@google.com.

Thank you for reporting this issue. For us to further investigate this issue, please provide the following additional information:

What steps are needed to reproduce this issue? Frequency of occurrence?

Which Android build are you using? (e.g. OPP1.170223.012)

Which device did you use to reproduce this issue?

Can you confirm if this issue is reproducible on a Pixel/Nexus device?

Android bug report (to be captured after reproducing the issue)
For steps to capture a bug report, please refer: https://developer.android.com/studio/debug/bug-report#bugreportdevice

Alternate method
Navigate to "Developer options", ensure "USB debugging" is enabled, then enable "Bug report shortcut". Capture bug report by holding the power button and selecting the "Take bug report" o

Note: Please upload the bug report and screenshot to google drive and share the folder to android-bugreport@google.com, then share the link here.

xi...@xiaomi.corp-partner.google.com <xi...@xiaomi.corp-partner.google.com> #3

Thank you for your feedback.

1. The problem did not reproduce the scene, and the frequency of occurrence: Development version -1/107, stable version 1/25.
2. Xiaomi M12 project: M12-V14.0.23.5.26. TMLCNXM version: Android 13.
3. Currently, Pixel/Nexus devices have not been replicated. Looking forward to your company's next suggestion, thank you!

vi...@google.com <vi...@google.com> #4

We have shared this with our product and engineering team and will update this issue with more information as it becomes available.

xi...@xiaomi.corp-partner.google.com <xi...@xiaomi.corp-partner.google.com> #5

Recently, the above situation has reappeared. Please help to take a look, thank you
As above, a null pointer was reported at the same position:

```
template <class _Tp, class _Hash, class _Equal, class _Alloc>
template <class _Key, class _Args>
pair<typename __hash_table<_Tp, _Hash, _Equal, _Alloc>::iterator, bool>
__hash_table<_Tp, _Hash, _Equal, _Alloc>::__emplace_unique_key_args(_Key const& __k, _Args& __args)
#endif
{

  size_t __hash = hash_function()(__k);
  size_type __bc = bucket_count();
  bool __inserted = false;
  __next_pointer __nd;
  size_t __chash;
  if (__bc != 0)
  {
    __chash = __constrain_hash(__hash, __bc);
    __nd = __bucket_list_[__chash];               //Final error position
    if (__nd != nullptr)
    {
      for (__nd = __nd->__next_; __nd != nullptr &&
        (__nd->__hash() == __hash || __constrain_hash(__nd->__hash(), __bc) == __chash);
                                        __nd = __nd->__next_)
      {
        if (key_eq()(__nd->__upcast()->__value_, __k))
          goto __done;
      }
    }
  }
```

Can locking be used to solve the problem? Or can you provide other solutions?

```
  ...
  size_t __chash;
  std::mutex mtx;
  if (__bc != 0)
  {
    __chash = __constrain_hash(__hash, __bc);
    __nd = __bucket_list_[__chash];
    if (__nd != nullptr)
    {
      for (__nd = __nd->__next_; __nd != nullptr &&
        (__nd->__hash() == __hash || __constrain_hash(__nd->__hash(), __bc) == __chash);
                                        __nd = __nd->__next_)
      {
        std::lock_guard<std::mutex> lock(mtx);
        if (key_eq()(__nd->__upcast()->__value_, __k))
          goto __done;
      }
    }
  }
```

deleted
0 B

vi...@google.com <vi...@google.com> #6

Thanks for reporting this issue!

The crash is usually due to unsychronized access to the Transaction object. The call seems to be coming from code that is not in our internal framework .

Specifically, `AppWindowAnimatorHelper#destoryMiuiActivityThumbnailLeash` does not exist. Please note that the SurfaceControl#Transaction objects do not have any built in synchroni

xi...@xiaomi.corp-partner.google.com <xi...@xiaomi.corp-partner.google.com> #7

Thank you very much for your answer

vi...@google.com <vi...@google.com> #8

Status: Won't Fix (Intended Behavior)

Based on the above comment we are closing this issue, thanks!