NPE in Robolectric's NativeObjRegistry.getNativeObject due to LayoutIncludeDetector calling getName on closed XmlBlock

Obsolete Bug P1 + Add Hotlist

STATUS UPDATE No update yet. Edit

DESCRIPTION al...@google.com created issue #1

Unclear if this affects real apps or just Robolectric tests, but we'll need to fix it regardless:

```
Caused by: java.lang.NullPointerException: Could not find object with nativeId: 0. Currently registered ids: [13, 14, 15, 16, 17, 18, 22, 23, 27, 28, 29, 33,
        at org.robolectric.res.android.NativeObjRegistry.getNativeObject(NativeObjRegistry.java:141)
        at org.robolectric.shadows.ShadowXmlBlock.getResXMLParser(ShadowXmlBlock.java:349)
        at org.robolectric.shadows.ShadowXmlBlock.nativeGetName(ShadowXmlBlock.java:157)
        at org.robolectric.shadows.ShadowXmlBlock.nativeGetName(ShadowXmlBlock.java:152)
        at android.content.res.XmlBlock.nativeGetName(XmlBlock.java)
        at android.content.res.XmlBlock$Parser.getName(XmlBlock.java:174)
        at android.support.v7.app.LayoutIncludeDetector.shouldInheritContext(LayoutIncludeDetector.java:78)
        at android.support.v7.app.LayoutIncludeDetector.detect(LayoutIncludeDetector.java:56)
        at android.support.v7.app.AppCompatDelegateImpl.createView(AppCompatDelegateImpl.java:1550)
        at android.support.v7.app.AppCompatDelegateImpl.onCreateView(AppCompatDelegateImpl.java:1614)
```

In the platform code, it's closing (and destroying the native object) when it reaches the end of an XML document:

```
            if (ev == END_DOCUMENT) {
                // Automatically close the parse when we reach the end of
                // a document, since the standard XmlPullParser interface
                // doesn't have such an API so most clients will leave us
                // dangling.
                close();
            }
```

As a result, calling any method that touches the native object (like `getName`) will fail:

```
        public String getName() {
            int id = nativeGetName(mParseState);
            return id >= 0 ? mStrings.get(id).toString() : null;
        }
```

However, we're not seeing it fail with an NPE when running against a real device. And from the Robolectric code, it looks a `null` native object is a valid `END_DOCUMENT` state:

```
  protected static int nativeNext(long state) throws XmlPullParserException {
    ResXMLParser st = getResXMLParser(state);
    if (st == null) {
      return ResXMLParser.event_code_t.END_DOCUMENT;
    }
```

✓ Links (3)

Links (3)

"I filed an issue on the Robolectric Github tracker: https://github.com/robolectric/robolectric/issues/6292"
"From XmlBlock.getName: https://source.corp.google.com/android/frameworks/base/core/java/android/content/res/XmlBlock.java;l=196-199?q=xmlblock&ss=piper%2FGoogle..."
"https://source.corp.google.com/android/frameworks/base/core/jni/android_util_X... "

COMMENTS

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

Robolectric's `ShadowXmlBlock.getResXmlParser` is calling the wrong method on `NativeObjRegistry`:

```
  /** Retrieve the native object for given id. Throws if object with that id cannot be found */
  public synchronized T getNativeObject(long nativeId) {
    ...
  }
```

```
  /**
   * Similar to {@link #getNativeObject(long)} but returns null if object with given id cannot be
   * found.
   */
  public synchronized T peekNativeObject(long nativeId) {
    ...
  }
```

It should be calling `peekNativeObject` so that it doesn't crash on `null` (e.g. the object has been released).

ho...@google.com <ho...@google.com> #3

Thanks for digging into this alanv@. I'll try to come up with a reduced test case to understand the issue better. Also should this be transferred to `Mobile > Tools > Robolectric`?

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

I filed an issue on the Robolectric Github tracker: https://github.com/robolectric/robolectric/issues/6292

I'm planning to use this bug to track a workaround, if we need one.

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

Re: Reduced test case, it should be enough to create an XML parser on an arbitrary XML file, run it until it hits `END_DOCUMENT`, and then call `getName` on it. You could also create an XML pars

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

Ah, I see the problem.

Maybe the Robolectric implementation should follow the Android one:

From XmlBlock.getName: https://source.corp.google.com/android/frameworks/base/core/java/android/content/res/XmlBlock.java;l=196-199?q=xmlblock&ss=piper%2FGoogle%2Fandroid

```
        public String getName() {
            int id = nativeGetName(mParseState);
            return id >= 0 ? mStrings.get(id).toString() : null;
        }
```

ho...@google.com <ho...@google.com> #7

And it seems like the expectation is that nativeGetName returns -1 if the parser has been destroyed:

https://source.corp.google.com/android/frameworks/base/core/jni/android_util_XmlBlock.cpp;l=140-149

```
static jint android_content_XmlBlock_nativeGetName(JNIEnv* env, jobject clazz,
                                                   jlong token)
{
    ResXMLParser* st = reinterpret_cast<ResXMLParser*>(token);
    if (st == NULL) {
        return -1;
    }

    return static_cast<jint>(st->getElementNameID());
}
```

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

Here's what I found sort of odd:

```
  @Implementation(minSdk = VERSION_CODES.LOLLIPOP)
  protected static int nativeNext(long state) throws XmlPullParserException {
    ResXMLParser st = getResXMLParser(state);
    if (st == null) {
      return ResXMLParser.event_code_t.END_DOCUMENT;
    }
```

Clearly someone expected `getResXMLParser` to return `null` at `END_OF_DOCUMENT`, as it does in the platform.

```
  private static ResXMLParser getResXMLParser(long state) {
    return Registries.NATIVE_RES_XML_PARSERS.getNativeObject(state);
  }
```

But this can never return `null` because `getNativeObject` either returns non-`null` or crashes with NPE.

So, yeah, I'd expect all of the native methods in Robolectric to be consistent with the platform and return a default (e.g. `-1` value when `peekNativeObject` returns `null`.

al...@google.com <al...@google.com> #9

> as it does in the platform.

Or rather, for the native object to be dealloc'ed.

Especially weird that `nativeNext` handles this case, but none of the other `native` methods in `ShadowXmlBlock` do.

al...@google.com <al...@google.com>

Status: Won't Fix (Obsolete)