


Infeasible

Feature Request

P3

+ Add Hotlist

 STATUS UPDATE

No update yet.

Edit

 DESCRIPTION ha...@gmail.com created issue [#1](#)

In our game engine, we need the name of the native library used by `GameActivity`. To get hold of this, we use complicated JNI - reading manifest meta data - accomplishing the very same `PackageManager` thing you are already doing. It makes sense to store this value in an accessible field in `GameActivity` or, even better, in its native counterpart. Ideally, we should be able to read it in the same way we can read `internalDataPath`.

Sample code

```
extern "C" const char *GetInternalStoragePath()
{
    return engine.app->activity->internalDataPath; // Nice!
}

extern "C" const char *GetLibraryName()
{
    // This would be useful! Too bad we need some 100 lines of JNI... :(
    return engine.app->activity->libName;
}
```

Motivation

For a multi-platform engine, you need a common way to handle command line arguments. On `Linux` and `Windows`, the first argument is the process name. This can be used to pick matching code which is more or less equivalent. Thus, it should be exposed.

PS.

You could also consider having an additional meta-data tag as in our game engine (we have a different namespace, obviously):

```
<meta-data android:name="android.app.lib_name" android:value="funGame" />
<meta-data android:name="android.app.arguments" android:value="{gameArguments}" />
```


Allowing for this (extremely useful) API in the engine:

```
extern "C" const char *GetArguments()
{
    return engine.app->activity->arguments;
}
```

Comments

- ✓

Mentioned issues (1)
- ✓

Links (4)
- 

Mentioned issues (1)
- P3

C++shared required for static versions of `GameActivity` and `Paddleboat`? "All in all, you can close this issue. [Issue 266535057](#) is much more important!"


↔

Links (4)

"...eed the name of the native library used by `GameActivity`. To get hold of this, we use complicated JNI - reading manifest meta data - accomplishing the very same `PackageManager` thing you are already doing. If you need the process name you can excess it like explained here: <https://stackoverflow.com/questions/19631894/is-there-a-way-to-get-current-process-name-in-android>"


"...g you came up with. It defaults to "main" and then you have some logic where you can customize this from within the manifest as seen [↔ here](#)."

"In [↔ orx](#) we decided to use a custom metadata tag, with the `android.app.lib_name` as fallback value. It's a nice solution, but we can never guarantee its correctness. Reading your code, `GameActivity`

- COMMENTS
- 

vr...@google.com

<vr...@google.com>

Reassigned to ar...@google.com.
- 

ar...@google.com


<ar...@google.com>

[#2](#)


we need the name of the native library used by `GameActivity`

What do you mean by libName? File name? File path? Library name like "GameActivity"?

Message last modified on Sep 1, 2023 11:44PM

 **ar...@google.com** <ar...@google.com> [#3](#)


If you need the process name you can excess it like explained here: <https://stackoverflow.com/questions/19631894/is-there-a-way-to-get-current-process-name-in-android>

 **ha...@gmail.com** <ha...@gmail.com> [#4](#)

The `libName` is something you came up with. It defaults to "main" and then you have some logic where you can customize this from within the manifest as seen [↔ here](#).


In [↔ orx](#) we decided to use a custom metadata tag, with the `android.app.lib_name` as *fallback* value. It's a nice solution, but we can never guarantee its correctness. Reading your code, G, this issue.

In native code, we cannot repeat the code you have in `GameActivity`. Also, there is no virtual method where we could get hold of the value. I.e. `protected void onLibraryLoaded(string`
All in all, I think we can live without the knowledge of the library name. The custom metadata solution described above solves our problem. But why not add an `onLibraryLoaded()` method?

 **ar...@google.com** <ar...@google.com> [#5](#)

Btw, I took a look at ORX and it looks great! Could you give me an example of a possible usage of `onLibraryLoaded`? Why must you know the name of the native library file if it already has b


Message last modified on Sep 4, 2023 09:18PM

 **ha...@gmail.com** <ha...@gmail.com> [#6](#)

Thanks. It's a masterpiece indeed. (The Android integration is pretty good too.)

The way `orx` was designed, it reads an `.ini` configuration file with the same name as the executable (library name). For different setups/flavors, we wanted to use different names of the lib little need for the library name. That's where we are now.

So to answer your question; I don't think we need this feature anymore. At least not now. If we ever need to read the library name again, we could reconsider the `dladdr()` API even though w
All in all, you can close this issue. Issue 266535057 is much more important!

 **ar...@google.com** <ar...@google.com>

Status: Won't Fix (Infeasible)