



How to launch Khronos OpenGL ES conformance tests



Contents

1. How to launch Khronos OpenGL ES conformance tests	3
--	---



How to launch Khronos OpenGL ES conformance tests

Stable: 03.10.2019 - 13:22 / Revision: 26.09.2019 - 12:54

Contents

1 Introduction	3
2 How to install khronos-cts	3
2.1 Building khronos-cts	3
2.2 Installing khronos-cts on the target board	3
3 How to execute khronos-cts	4
3.1 Running test suite	4
3.2 Running a single/group of test(s)	4
4 Testing verdict example	5
5 References	5

1 Introduction

The purpose of this article is to describe how to build and execute the **Khronos OpenGL ES 2.0 Conformance Tests**.

These tests are provided by the Khronos Group ^[1]. They are available as open source from the **Khronos CTS GitHub source repo** ^[2].

For a detailed description of the Khronos OpenGL ES2.0 CTS, please refer to the **openglcts README.md** ^[3] file.

2 How to install khronos-cts

2.1 Building khronos-cts

Execute the following command in the OpenSTLinux build environment:

```
PC $> bitbake khronos-cts
```

2.2 Installing khronos-cts on the target board

Execute the following command in the OpenSTLinux build environment:

```
PC $> scp tmp*/deploy/deb/*neon*/khronos-cts_opengl-cts-<CTS Release>.deb  
root@<IPBOARD>:<SomewhereInTheBoard>
```



But default dpkg will extract within /home/root/.

This debian package is rather huge. For instance opengl-cts-4.6.0 is around 115MB. If rootfs space is at stake this is better to unpack the debian somewhere else by appending instdir=<SomewhereElse> to the above command

3 How to execute khronos-cts

3.1 Running test suite

Execute the following commands on the target board:

```
Board $> cd /home/root
Board $> ./cts-runner --type=es2 1>results.txt
[    1] EGL: enable default configs for conformance test
...
```



The test execution can take several hours.



The ***free*** system RAM amount required to complete this test suite is around 500MB. Under out of memory occurs.



3.2 Running a single/group of test(s)

```
Board $> cd /home/root
Board $> glcts.exe --deqp-case=dEQP-EGL.functional.*
```



4 Testing verdict example

```
Board $>
...
Test run totals:
  Passed:      13786/13957 (98.8%)
  Failed:      11/13957 (0.1%)
  Not supported: 131/13957 (0.9%)
  Warnings:    29/13957 (0.2%)
219/220 sessions passed, conformance test FAILED
```

In order to interpret result, please look at "Understanding the result"^[4]

Even more there are several tools available to process those test logs^[5]

5 References

- <https://www.khronos.org/>
- <https://github.com/KhronosGroup/VK-GL-CTS>
- <https://github.com/KhronosGroup/VK-GL-CTS/blob/master/external/openglcts/README.md#introduction>
- <https://github.com/KhronosGroup/VK-GL-CTS/tree/master/external/openglcts#understanding-the-results>
- <https://github.com/KhronosGroup/VK-GL-CTS/tree/master/external/openglcts#test-logs>

Open Graphics Library (See <http://www.opengl.org/> for more details)

Compatibility Test Suite (Android specific) or Clear to send (in UART context)

Khronos Native Platform Graphics Interface (See <http://www.khronos.org/egl/> for more details)

Random Access Memory (Early computer memories generally had serial access. Memories where any given address can be accessed when desired were then called "random access" to distinguish them from the memories where contents can only be accessed in a fixed order. The term is used today for volatile random-access semiconductor memories.)