

Android CameraService提供的强大Tag Monitor机制，太好用了！

在Camera Debug过程中，我们经常会遇到如下问题：

1. Camera App在某一帧设置的参数值是什么？
2. Camera HAL在某一帧返回的参数值是什么？

本文会从Camera Native Framework的角度，使用watch命令带大家轻松解决上面的问题

watch 命令

启动Tag Monitor

在**打开Camera前或后**执行如下命令，开始monitor 某个/某些Tags

```
adb shell cmd media.camera watch start -m <tags> [-c <clients>]
```

其中，

- -m 指定要monitor的Tags，tag之间通过逗号分隔
- -c 可选，指定要monitor那个/哪些Client（也就是Camera App）的Tag，Client之间用都好分隔，如果不指定或指定为all，则monitor所有已经打开的clients

举例：

```
adb shell cmd media.camera watch start \  
-m android.control.effectMode,android.control.aeMode \  
-c com.google.android.GoogleCamera,com.android.chrome
```

注：

- Tag也可以带预定义的名称，默认有一个名称: 3a，会monitor如下Tags（大家可以根据需求修改源码添加更多的预定义名称）

```
const char* TagMonitor::k3aTags =  
    "android.control.aeMode, android.control.afMode, android.control.awbMode,"  
    "android.control.aeState, android.control.afState, android.control.awbState,"  
    "android.control.aePrecaptureTrigger, android.control.afTrigger,"  
    "android.control.aeRegions, android.control.awbRegions, android.control.afRegions,"  
    "android.control.aeExposureCompensation, android.control.aeLock, android.control.awbLock,"  
    "android.control.aeAntibandingMode, android.control.aeTargetFpsRange,"  
    "android.control.effectMode, android.control.mode, android.control.sceneMode,"  
    "android.control.videoStabilizationMode";
```

Dump Tag Monitor结果

执行如下命令可以dump monitor的结果，close的Camera的tag monitor也能dump到

```
adb shell cmd media.camera watch dump
```

举例，我们Monitor FD模式的Tag:

```
adb shell cmd media.camera watch start -m android.statistics.faceDetectMode
```

操作Camera后，执行watch dump，会输出如下信息

```
Cached @ 06-13 09:21:06  
0:com.android.messaging f0:67654301839288ns: REQ: output stream ids: 0  
0:com.android.messaging f0:67654301839288ns: REQ:android.statistics.faceDetectMode: [OFF]  
0:com.android.messaging f0:743751184000ns: RES:android.statistics.faceDetectMode: [OFF]
```

其中，

- Cached @ 06-13 09:21:06 表示Cache的时间点
- 0:com.android.messaging，0表示CameraID为0，com.android.messaging 是打开Camera 0的Camera App
- f0:67654301839288ns，f0表示framenummer 0，67654301839288是存储这个event的时间点
- output stream ids，表示这个Metadata带有哪些Stream ID的buffer
- REQ表示CaptureRequest
- RES表示CaptureResult

实时Dump Tag Monitor结果

有时候我们需要实时知道Tag 值的变化，Google也想到了，watch支持在线实时dump tag，执行：

```
adb shell cmd media.camera watch live [-n refresh_interval_ms]
```

其中，refresh_interval_ms是刷新闻隔，如果不设置默认值是1000ms（Tag值有变化输出才会更新），敲回车后会退出实时dump

```

$ adb shell # cmd media.camera watch live -n 33
Press return to exit...

0:com.android.messaging f0:77590507280873ns: REQ: output stream ids: 0
0:com.android.messaging f0:77590507280873ns: REQ:android.sensor.exposureTime: [100000 ]
0:com.android.messaging f0:10679969702000ns: RES:android.sensor.exposureTime: [30000000 ]
0:com.android.messaging f2:10680036899000ns: RES:android.sensor.exposureTime: [20000000 ]
0:com.android.messaging f4:10680104084000ns: RES:android.sensor.exposureTime: [30000000 ]
0:com.android.messaging f41:10681353578000ns: RES:android.sensor.exposureTime: [40000000 ]
0:com.android.messaging f44:10681467344000ns: RES:android.sensor.exposureTime: [30000000 ]
0:com.android.messaging f45:10681500915000ns: RES:android.sensor.exposureTime: [20000000 ]
0:com.android.messaging f49:10681635302000ns: RES:android.sensor.exposureTime: [30000000 ]
0:com.android.messaging f64:10682145732000ns: RES:android.sensor.exposureTime: [40000000 ]
0:com.android.messaging f65:10682195810000ns: RES:android.sensor.exposureTime: [50000000 ]
0:com.android.messaging f71:10682486230000ns: RES:android.sensor.exposureTime: [40000000 ]
0:com.android.messaging f76:10682680135000ns: RES:android.sensor.exposureTime: [30000000 ]
0:com.android.messaging f79:10682781006000ns: RES:android.sensor.exposureTime: [20000000 ]
0:com.android.messaging f82:10682881706000ns: RES:android.sensor.exposureTime: [30000000 ]
0:com.android.messaging f87:10683049672000ns: RES:android.sensor.exposureTime: [20000000 ]
0:com.android.messaging f101:10683519996000ns: RES:android.sensor.exposureTime: [30000000 ]
0:com.android.messaging f102:10683560071000ns: RES:android.sensor.exposureTime: [40000000 ]
0:com.android.messaging f106:10683713913000ns: RES:android.sensor.exposureTime: [30000000 ]
0:com.android.messaging f107:10683747581000ns: RES:android.sensor.exposureTime: [20000000 ]
0:com.android.messaging f113:10683949091000ns: RES:android.sensor.exposureTime: [30000000 ]
0:com.android.messaging f117:10684083432000ns: RES:android.sensor.exposureTime: [20000000 ]
0:com.android.messaging f122:10684251406000ns: RES:android.sensor.exposureTime: [30000000 ]
0:com.android.messaging f125:10684358713000ns: RES:android.sensor.exposureTime: [40000000 ]
0:com.android.messaging f127:10684432340000ns: RES:android.sensor.exposureTime: [30000000 ]
```

清除Tag Monito缓存的Dump

如果Dump的内容太多不便于查看，也可以执行如下命令清楚掉缓存的Dump

```
adb shell cmd media.camera watch clear
```

停止Tag Monitor的Dump

Monitor后，记得要停止watch

```
adb shell cmd media.camera watch stop
```

dumpsys方式使用Tag Monitor

如果你觉得使用watch命令要敲太多命令，可以使用dumpsys来使用Tag Monitor

```
adb shell dumpsys media.camera -m 3a | grep -A50 Monitored
```

其中，

- -m 表示monitor哪个/哪些Tag
- 结果会呈现在dumpsys media.camera的Monitored后面

```

dumpsys media.camera -m android.statistics.faceRectangles | grep -A50 Monitored
Monitored tag event log:
f514:2335265512000ns: RES:android.statistics.faceRectangles: [820 460 2140 1560 ]
f191:2324324453000ns: RES:android.statistics.faceRectangles: [680 540 2000 1640 ]
f188:2324222838000ns: RES:android.statistics.faceRectangles: [690 470 2010 1680 ]
f90:2320903261000ns: RES:android.statistics.faceRectangles: [610 460 2070 1680 ]
f28:2318803116000ns: RES:android.statistics.faceRectangles: [760 400 2220 1620 ]
```

借助linux的watch命令，dumpsys 也可以实时打印Dump结果，比如：

```
watch -n 1 -c 'adb shell dumpsys media.camera -m 3a | grep -A50 Monitored'
```

其中，

- -n 1: 这个选项告诉watch每1秒（1秒的时间间隔）执行一次命令。如果没有这个选项，默认时间间隔是2秒。
- -c: 这个选项表示在每次执行命令前清除屏幕，使得每次watch执行命令时，你都能看到最新的输出，而不是累积的输出。