

Android中打印代码的调用层次

标签: [androidexceptionjavanull](#)

2012-08-08 14:27 1310人阅读 [评论\(0\)](#) [收藏](#) [举报](#)

分类:

[android \(33\)](#)

<http://hubingforever.blog.163.com/blog/static/171040579201172282210766/>

在我们调试Android代码的时候,有些时候需要取得当前代码的调用层次(调用栈)。

本人现知3种方法,一种是java传统的方法,另一种是Android提供的方法,还有第三种需借助于Exception或Throwable。

第一种,通过getStackTrace()

这种是java的传统方法,即在你的代码处,通过Thread.currentThread().getStackTrace()得到一个StackTraceElement数组,然后打印该数组。

示例1:

```
StackTraceElement st[]= Thread.currentThread().getStackTrace();
```

```
for(int i=0;i<st.length;i++)
```

```
System.out.println(i+":"+st[i]);
```

这样你就把此处代码此时的调用层次(调用栈)打印出来了。

第二种,使用android.util.Log.wtf()

我们可以直接使用android提供的android.util.Log.wtf()系列函数来输出一个日志.在输出日志的同时,它会把此处代码此时的执行路径(调用栈)打印出来。

示例2:

```
Log.wtf(tag, "this a fake erro");
```

第三种,借助Exception

示例3:

```
try{
    cm.setPrimaryClip(ClipData.newRawUri(null, Uri.parse(text.toString())));
}catch(Exception e)
{
    e.printStackTrace();
}
```

示例4:

```
try{
    .....
}catch(Exception e)
{
    Log.wtf("robin", e);
}
```

注意我们其实可以直接构建Exception或Throwable。