每天的点点滴滴

**问题 i：**

**问题描述：**

**错误表现：**

**错误原因:**

**解决方案：**

**举一反三：**

**知识点**

2015/6/18

1. **EHB Testing的两大难点及小问题**
   1. **Soot的问题。**转换后的App含有错误，运行出错。
   2. **调用CallBack的问题。**多个参数的CallBack，模拟其输入参数。

说明：一个callback的多个输入， callback有多条path，是否测试每条path？

注意：多个参数，输入多个测试用例

* 1. 搜索setOnClickListener或callBack。
  2. Xml的CallBack。

**问题2**

**问题描述：**AXmlPrint.jar与Android.jar冲突。

**错误表现：**

Exception in thread "main" java.lang.IllegalAccessError: tried to access method test.AXMLPrinter.getAttributeValue(Landroid/content/res/AXmlResourceParser;I)Ljava/lang/String; from class soot.Scene

at soot.Scene.getAndroidJarPath(Scene.java:349)

at soot.Scene.defaultClassPath(Scene.java:472)

at soot.Scene.getSootClassPath(Scene.java:223)

at soot.SootResolver.<init>(SootResolver.java:81)

**错误原因:** AXmlPrint.jar与Android.jar命名冲突

**解决方案：**引入AXmlPrint.jar源码，并将包名替换成非andorid开头

**举一反三：**

1. **包名切勿以android, java开头**
2. **对于包名冲突的情况，引入源代码是一种好的解决方案。**

**问题3**

**问题描述：测试RingDroid时，界面A中的事件被第一次触发，没有报错，第二次被调用，报错**

**错误表现：**

V/EVENT ( 6202): Bugs Detected, InvocationTargetException!!! Message: null Cau

sed by android.view.WindowManager$BadTokenException: Unable to add window -- tok

en android.os.BinderProxy@b2d99558 is not valid; is your activity running?

W/System.err( 6202): android.view.WindowManager$BadTokenException: Unable to add

window -- token android.os.BinderProxy@b2d99558 is not valid; is your activity

running?

W/WindowManager( 415): Attempted to add application window with unknown token T

oken{b32e33d8 null}. Aborting.

W/System.err( 6202): at android.view.ViewRootImpl.setView(ViewRootImpl.java:5

36)

W/System.err( 6202): at android.view.WindowManagerGlobal.addView(WindowManage

rGlobal.java:259)

W/System.err( 6202): at android.view.WindowManagerImpl.addView(WindowManagerI

mpl.java:69)

W/System.err( 6202): at android.app.Dialog.show(Dialog.java:286)

W/System.err( 6202): at android.app.Activity.showDialog(Activity.java:3125)

W/System.err( 6202): at android.app.Activity.showDialog(Activity.java:3075)

W/System.err( 6202): at com.irahul.worldclock.TimeZoneEdit$1.onClick(TimeZone

**错误原因: 未解决**

**解决方案：**

**举一反三：**

**知识点1**

**onClick()是等待调用，而非及时调用，留作调用**

**知识点2**

**自定义的view, 通过setOnClickListener(listener)来监听事件，与view.setOnClickLisnter(listener)监听方式并无不同。**

**setOnClickListener(listener)= this.setonClickListener(listener);**

**问题4 2015/7/7**

**问题描述：**Tool工具instrument Dubatterysaver，DespicableMe等App时报错。

**错误表现：**Exception in thread "main" java.lang.Error: unknown SpecialInvokeExpr (no call to constructor, super or private method): specialinvoke activity.<android.app.Activity: boolean onCreateOptionsMenu(android.view.Menu)>(menu)

**出错原因：**AndroidManifest定义为Activity的类，并没有实现Activity。

**解决方案：**判断从AndroidManifest读取的Activity是否是真的Activity，若不是则抛出ClassIsNotActivity异常。

**举一反三：**

**知识点3**

**Android App的系统事件是由系统触发，不能被应用App触发。**

**从内部调用这些事件，事件最终被onReceiver(), onService()捕获**

**注册广播监听有两种机制：**

1. **静态监听，在Manifest中加入Intent-filter等**
2. **动态监听，在运行时刻加入context.register(broadCastReceiver,intentFilter);**

**注册服务监听也有两种机制：**

1. **静态监听，在Manifest中加入Intent-filter等**
2. **动态监听，运行时刻，LocationManager.** **addGpsStatusListener(Listener l)**

**总之： 对于Android App，App的状态改变，都有可能执行操作，因而所有事件都需要考虑，不管事件的种类。**

**代码执行覆盖率：x随着事件的触发而提高。**

**1%>5%>10%>…50%>70%>90%>100%**

**知识点4**

**INSTALL\_FAILED\_CONTAINER\_ERROR**

**问题5 2015/7/9**

**07-08 04:09:18.685: E/AndroidRuntime(2330): java.lang.SecurityException: Permission Denial: not allowed to send broadcast android.intent.action.BOOT\_COMPLETED from pid=2330, uid=10063**

**问题6 2015/7/13**

**问题描述：已经将class添加进App, 但依然报java.lang.NoClassDefFoundError**

**错误表现：E/AndroidRuntime( 1407): java.lang.NoClassDefFoundError:com.app.test.event.SystemEventHandler**

**错误原因: 虽然SystemEventHandler被设为applicationClass,但其引用的其他类却设为ApplicationClass，在运行时找不到这些引用类，因此报错。**

**解决方案：将添加的class所引用的所有class都添加进App。**

**举一反三：**

**知识点5**

**System.out.println((s1.equals(s2))+" "+(s1==s2)); true false**

**知识点6**

**println((s1.equals(s2))+" "+( s1.intern.==(s2.intern)); true true**

**intern() 将String对象转换成常量**

**知识点7**

Intent Filter android:pathPattern =”.\*\\.pdf”

在path中 “.”表示任意字符，”\*”表示可以出现多次 “[\\.](file:///\\.)”表示转义”.” 两次转义的原因是 1次是Xml中的转义，另一次是pathPattern中的转义.因而上述表达的意思是\*.pdf

**知识点8**

**Android无法用发射获得onCreate(),onStart()等生命周期方法，也不能获得onListItemClick();**

**知识点9**

**新版的Soot，可能会更改部分功能，目前遇到被更改的功能有：**

**Value invoker = invokeExpr.getUseBoxes().get(0).getValue();**

**以前是：返回invoker**

**现在是：返回第一个参数**

**问题7**

**问题描述：App启动时直接崩溃**

**错误表现：java.lang.VerifyError: cri/sanity/util/Alert**

**错误原因: 静态方法调用了非静态方法，原先doReflect设置成非静态方法出错**

**解决方案：将DoReflect改成静态方法即可**

**举一反三：错误之处在Alert类，则说明错误就在Alert类中。只有明确错误所在的具体类，才能继续搜寻问题所在**

**问题8**

**问题描述：app ToDex阶段报错**

**错误表现：Paramter local not found**

**错误原因: 将DoReflect修改为static后，未将base = this去掉**

**解决方案：去掉base=this**

**举一反三：ToDex阶段的报错信息，并不能锁定bug，只有全面了解程序，才能将这类bug找到。**

**知识点10**

**去掉Transforming com.flurry.android.Flog... 采用正则表达式**

**Transforming [.a-zA-Z0-9$.]\***

**知识点11**

**transfer int to long**

**String s = String.valueOf(index);**

**Long lon = Long.valueOf(s);**

**long l = getListAdapter();**

**知识点12**

**Add Insert(count) to method**

**Count值在静态分析后是一个常量，而非变量。程序执行时，也是常量**

**问题9**

**问题描述：重打包的freenovel.apk启动后，强制退出**

**错误表现：app加载成功后，弹出对话框，强制退出**

**错误原因: 该app运行时，检测签名是否正确。在com.dzbook.activity.MainActivity中存在方法:**

private boolean isSignLegal(Context paramContext)

{

String str1 = getSign(paramContext);

String str2;

if (!TextUtils.isEmpty(str1))

{

str2 = c.a(str1);

ag.h("md5sign:" + str2);

String[] arrayOfString = this.releaseSign;

int i = arrayOfString.length;

for (int j = 0; j < i; j++) {

if (arrayOfString[j].equals(str2)) {

return true;

}

}

}

return true;

return this.debugSign.equals(str2);

}

**解决方案：去掉checkSignLegal()**

**举一反三：如果每个测试app都存在签名检测，则设置Map = <app, 检测方法>，以此来注释掉这些检测方法。**

**知识点13 ActionBar的使用**

**现在许多App都采用如下界面做为App的主界面，用到了ActionBar。**

Google自4.0之后引入了ActionBar取代Menu，希望所有的UI都能够在一个界面上展示，

并逐步削弱Menu和Back这两个实体按键的功能。



1. **ActionBar的设计：**

ActionBar的设计由两部分组成: 左上角的menu与右上角的menu。左上角的menu的设计包含

三个部分：theme，icon和title；右上角的menu可显示的menuitem由所剩的空间决定，上图

显示了3个item，如果超过三个item，超过的部分折叠在item3中。

1. **ActionBar的实现：**

a) Theme，icon和title的实现：

应用于整个App：

<application

android:name=*"MyApplication"*

android:allowBackup=*"true"*

android:icon=*"@drawable/ic\_launcher"*

android:label=*"@string/app\_name"*

android:theme=*"@style/AppTheme"*

/>

应用于单个activity:

<activity

android:name=*"com.example.qian.MainActivity"*

android:icon=*"@drawable/ic\_launcher"*

android:label=*"@string/app\_name"*

android:theme=*"@style/AppTheme"/*>

响应点击事件：

**public** **boolean** onOptionsItemSelected(MenuItem item) {

**switch** (item.getItemId()) {

**case** android.R.id.*home*:

Toast.*makeText*(**this**, "home is clicked", 1).show();

**break**;

**return** **super**.onOptionsItemSelected(item);

}

b) MenuItem的实现：

方式1： 使用MenuInflater实现

**public** **boolean** onCreateOptionsMenu(Menu menu) {

MenuInflater inflater = getMenuInflater();

inflater.inflate(R.menu.*main*, menu);

**return** **true**; }

方式2： 使用menu.addMenuItem()实现

**public** **boolean** onCreateOptionsMenu(Menu menu) {

menu.add("Save")

.setIcon(R.drawable.*ic\_compose\_inverse*)

.setShowAsAction(MenuItem.*SHOW\_AS\_ACTION\_IF\_ROOM*);

menu.add("Search")

.setShowAsAction(MenuItem.*SHOW\_AS\_ACTION\_IF\_ROOM* | MenuItem.*SHOW\_AS\_ACTION\_WITH\_TEXT*);

menu.add("Refresh")

.setIcon(R.drawable.*ic\_refresh\_inverse*)

.setShowAsAction(MenuItem.*SHOW\_AS\_ACTION\_IF\_ROOM* | MenuItem.*SHOW\_AS\_ACTION\_WITH\_TEXT*);

**return** **true**;}