# StrictMode

StrictMode最常用来捕捉应用程序的主线程，它将报告与线程及虚拟机相关的策略违例。一旦检测到策略违例（policy violation），你将获得警告，其包含了一个栈trace显示你的应用在何处发生违例。除了主线程，我们还可以在Handler，AsyncTask，AsyncQueryHandler，IntentService等API中使用StrictMode。

## 检查策略

StrictMode的线程策略主要用于检测磁盘IO和网络访问，而虚拟机策略主要用于检测内存泄漏现象。Android已经在磁盘IO访问和网络访问的代码中已经加入了StrictMode。当监视的线程发生策略的违例时，就可以获得警告，例如写入LogCat，显示一个对话框，闪下屏幕，写入DropBox日志文件，或让应用崩溃。最通常的做法是写入LogCat或让应用崩溃。下面的代码展示了如何使用StrictMode的检查策略：

|  |
| --- |
| public void onCreate() {  if (DEVELOPER\_MODE) {  StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder()  .detectDiskReads()  .detectDiskWrites()  .penaltyDialog()  .detectNetwork() // or .detectAll() for all detectable problems  .penaltyLog()  .build());  StrictMode.setVmPolicy(new StrictMode.VmPolicy.Builder()  .detectLeakedSqlLiteObjects()  .detectLeakedClosableObjects()  .penaltyLog()  .penaltyDeath()  .build());  }  super.onCreate();  } |

如果不指定检测函数，也可以用detectAll()来替代。penaltyLog()表示将警告输出到LogCat，你也可以使用其他或增加新的惩罚（penalty）函数，例如使用penaltyDeath()的话，一旦StrictMode消息被写到LogCat后应用就会崩溃。具体支持的监视方法见：<https://developer.android.com/reference/android/os/StrictMode.ThreadPolicy.Builder.html>与<https://developer.android.com/reference/android/os/StrictMode.VmPolicy.Builder.html>

## 使用方法

在正式版本中，我们并不希望使用StrictMode来让用户的应用因为一个警告而崩溃，所以在应用正式发布时，需要移出这些监视。你可以通过删除代码来实现，不过这里提供一个更好的方式来解决这个问题，即使用AndroidMainifest文件中的debuggable属性来实现，代码如下所示：

|  |
| --- |
| android:debuggable="true" |

在代码中，使用方法如下所示：

|  |
| --- |
| // Return if this application is not in debug mode  ApplicationInfo appInfo = context.getApplicationInfo();  int appFlags = appInfo.flags;  if ((appFlags & ApplicationInfo.FLAG\_DEBUGGABLE) != 0) {  // Do StrictMode setup here  StrictMode.setVmPolicy(new StrictMode.VmPolicy.Builder()  .detectLeakedSqlLiteObjects()  .penaltyLog()  .penaltyDeath()  .build());  } |

## StrictMode实例

我们在测试代码的主线程中去访问网络，这样就一定会触发StrictMode的线程监测，代码如下所示：

|  |
| --- |
| public class MainActivity extends Activity {  private HttpResponse httpResponse = null;  private HttpEntity httpEntity = null;  private TextView mTextView;  @Override  protected void onCreate(Bundle savedInstanceState) {  StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder()  .detectDiskReads()  .detectDiskWrites()  .detectNetwork()  .penaltyDialog()  .penaltyLog()  .build());  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  mTextView = (TextView) findViewById(R.id.text);  HttpGet httpGet = new HttpGet("http://www.baidu.com");  HttpClient httpClient = new DefaultHttpClient();  InputStream inputStream = null;  try {  httpResponse = httpClient.execute(httpGet);  httpEntity = httpResponse.getEntity();  if (httpResponse.getStatusLine().getStatusCode() == 200) {  inputStream = httpEntity.getContent();  BufferedReader reader =  new BufferedReader(new InputStreamReader(inputStream));  mTextView.setText(reader.readLine());  }  } catch (Exception e) {  e.printStackTrace();  } finally {  try {  inputStream.close();  } catch (IOException e) {  e.printStackTrace();  }  }  }  } |

运行代码，并将Log信息保存到本地，在Log中，我们可以搜索D/StrictMode(15454): StrictMode policy violation关键字。这里截取一段，如下所示：

|  |
| --- |
| D/StrictMode(15454): StrictMode policy violation; ~duration=461 ms:  android.os.StrictMode$StrictModeNetworkViolation: policy=55 violation=4  D/StrictMode(15454): at android.os.StrictMode$AndroidBlockGuardPolicy.onNetwork(StrictMode.java:1153)  D/StrictMode(15454): at libcore.io.BlockGuardOs.recvfrom(BlockGuardOs.java:249)  D/StrictMode(15454): at libcore.io.IoBridge.recvfrom(IoBridge.java:553)  D/StrictMode(15454): at java.net.PlainSocketImpl.read(PlainSocketImpl.java:485)  D/StrictMode(15454): at java.net.PlainSocketImpl.access$000(PlainSocketImpl.java:37)  D/StrictMode(15454): at java.net.PlainSocketImpl$PlainSocketInputStream.read(PlainSocketImpl.java:237)  D/StrictMode(15454): at org.apache.http.impl.io.AbstractSessionInputBuffer.fillBuffer(AbstractSessionInputBuffer.java:103)  D/StrictMode(15454): at org.apache.http.impl.io.AbstractSessionInputBuffer.read(AbstractSessionInputBuffer.java:134)  D/StrictMode(15454): at org.apache.http.impl.io.ChunkedInputStream.read(ChunkedInputStream.java:161)  D/StrictMode(15454): at org.apache.http.impl.io.ChunkedInputStream.read(ChunkedInputStream.java:175)  D/StrictMode(15454): at org.apache.http.impl.io.ChunkedInputStream.exhaustInputStream(ChunkedInputStream.java:289)  D/StrictMode(15454): at org.apache.http.impl.io.ChunkedInputStream.close(ChunkedInputStream.java:262)  D/StrictMode(15454): at org.apache.http.conn.BasicManagedEntity.streamClosed(BasicManagedEntity.java:179)  D/StrictMode(15454): at org.apache.http.conn.EofSensorInputStream.checkClose(EofSensorInputStream.java:266)  D/StrictMode(15454): at org.apache.http.conn.EofSensorInputStream.close(EofSensorInputStream.java:213)  D/StrictMode(15454): at com.imooc.strictmodetest.MainActivity.onCreate(MainActivity.java:53)  D/StrictMode(15454): at android.app.Activity.performCreate(Activity.java:5976)  D/StrictMode(15454): at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1105)  D/StrictMode(15454): at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2262)  D/StrictMode(15454): at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2371)  D/StrictMode(15454): at android.app.ActivityThread.access$800(ActivityThread.java:149)  D/StrictMode(15454): at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1289)  D/StrictMode(15454): at android.os.Handler.dispatchMessage(Handler.java:102)  D/StrictMode(15454): at android.os.Looper.loop(Looper.java:135)  D/StrictMode(15454): at android.app.ActivityThread.main(ActivityThread.java:5260)  D/StrictMode(15454): at java.lang.reflect.Method.invoke(Native Method)  D/StrictMode(15454): at java.lang.reflect.Method.invoke(Method.java:372)  D/StrictMode(15454): at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:898)  D/StrictMode(15454): at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:693) |

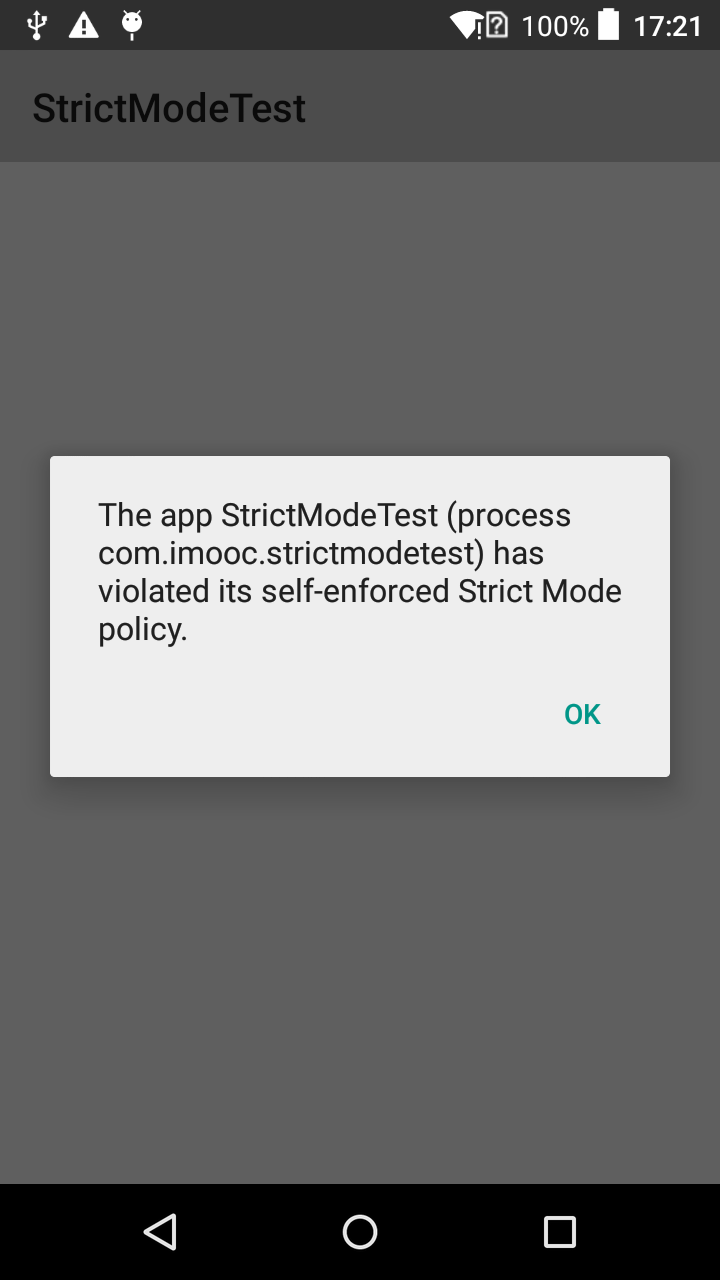
Log中可以显示出StrictMode提示的原因，通过这里的Trace Log我们就可以来找到优化的方法。除了在Logcat中查看StrictMode的日志信息，如果你使用了penaltyDropbox()方法，那么你还可以通过如下所示的命令来调用DropBoxManager观察StrictMode日志：

|  |
| --- |
| adb shell dumpsys dropbox data\_app\_strictmode --print |

输出日志文件如下所示：

|  |
| --- |
| ========================================  2015-03-09 17:47:14 data\_app\_strictmode (text, 2177 bytes)  Process: com.imooc.strictmodetest  Flags: 0x88be46  Package: com.imooc.strictmodetest v1 (1.0)  Build: TCL/idol347/idol347:5.0.2/LRX22G/1040:userdebug/release-keys  System-App: false  Uptime-Millis: 389545133  Loop-Violation-Number: 4  Duration-Millis: 520  android.os.StrictMode$StrictModeNetworkViolation: policy=183 violation=4  at android.os.StrictMode$AndroidBlockGuardPolicy.onNetwork(StrictMode.java:1153)  at java.net.InetAddress.lookupHostByName(InetAddress.java:418)  at java.net.InetAddress.getAllByNameImpl(InetAddress.java:252)  at java.net.InetAddress.getAllByName(InetAddress.java:215)  at org.apache.http.impl.conn.DefaultClientConnectionOperator.openConnection(DefaultClientConnectionOperator.java:137)  at org.apache.http.impl.conn.AbstractPoolEntry.open(AbstractPoolEntry.java:164)  at org.apache.http.impl.conn.AbstractPooledConnAdapter.open(AbstractPooledConnAdapter.java:119)  at org.apache.http.impl.client.DefaultRequestDirector.execute(DefaultRequestDirector.java:360)  at org.apache.http.impl.client.AbstractHttpClient.execute(AbstractHttpClient.java:555)  at org.apache.http.impl.client.AbstractHttpClient.execute(AbstractHttpClient.java:487)  at org.apache.http.impl.client.AbstractHttpClient.execute(AbstractHttpClient.java:465)  at com.imooc.strictmodetest.MainActivity.onCreate(MainActivity.java:43)  at android.app.Activity.performCreate(Activity.java:5976)  at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1105)  at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2262)  at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2371)  at android.app.ActivityThread.access$800(ActivityThread.java:149)  at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1289)  at android.os.Handler.dispatchMessage(Handler.java:102)  at android.os.Looper.loop(Looper.java:135)  at android.app.ActivityThread.main(ActivityThread.java:5260)  at java.lang.reflect.Method.invoke(Native Method)  at java.lang.reflect.Method.invoke(Method.java:372)  at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:898)  at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:693) |

同时，如果使用了penaltyDialog()方法，在应用中还会弹出如下所示的提示框：



## 暂停监测

如果在程序运行中无法避免的会违反StrictMode中的一些定义好的策略，而我们又希望能够暂时忽略这些策略的监视，我们可以使用permitXXXXX方法来暂停这些内容的监测，在做完需要忽略的监测之后，再起用监测，代码如下所示：

|  |
| --- |
| StrictMode.ThreadPolicy old = StrictMode.getThreadPolicy();  StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder(old)  .permitDiskWrites()  .build());  //doSomethingWriteToDisk();  StrictMode.setThreadPolicy(old); |