

Android Lint使用总结

一、Lint简介

Android Lint是在Android SDK Tools 16 提供的一套静态代码分析工具，对app中的代码规范带来了极大的方便。对内存泄漏、代码冗余、代码安全、国际化、代码规范等很多方面都能检测。

优化的内容主要包括以下几方面：

- Correctness：不够完美的编码，比如硬编码、使用过时 API 等；
- Performanc：对性能有影响的编码，比如：静态引用，循环引用等；
- Internationalization：国际化，直接使用汉字，没有使用资源引用等；
- Security：不安全的编码，比如在 WebView 中允许使用 JavaScript Interface 等；
- Usability：可用的，有更好的替换的，比如排版、图标格式建议.png格式 等；
- Accessibility：辅助选项，比如 ImageView 的 contentDescription 往往建议在属性中定义等；

Lint 检测代码的过程如下图所示：

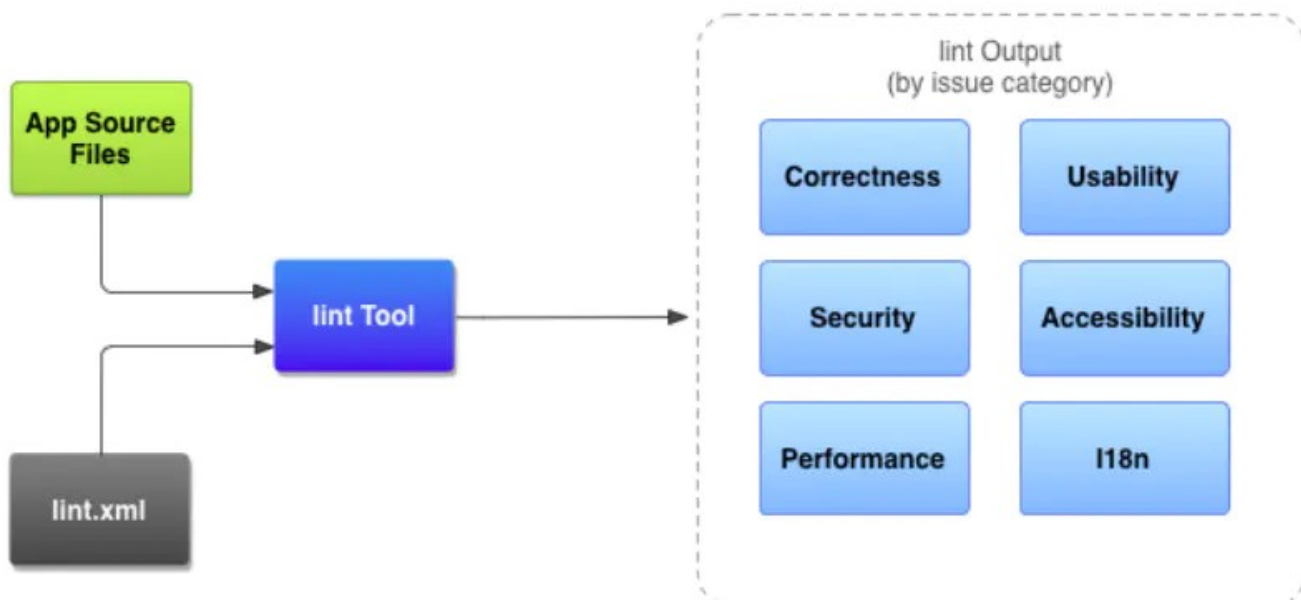
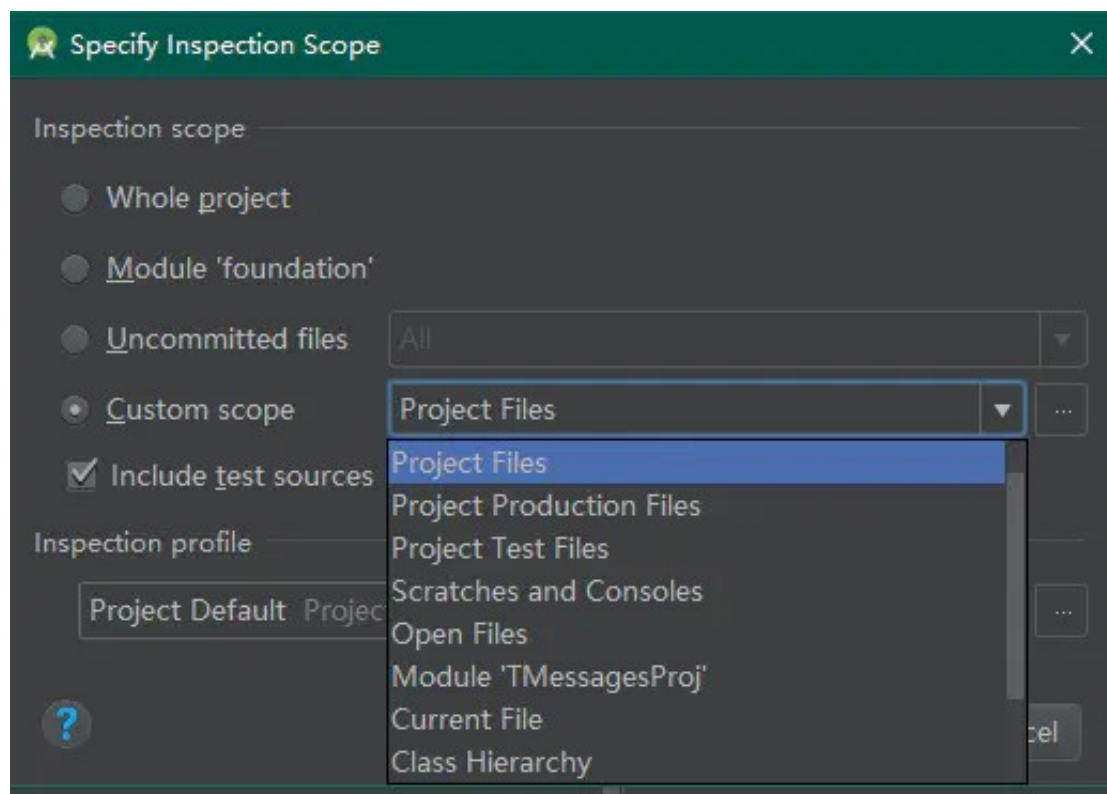


图 1. Lint 工具的代码扫描 workflow

二、使用方式

选择 Analyze -> Inspect Code, 打开 Lint 检查范围选择弹窗, 根据需要自行选择, 默认选择 Whole project 即可



- Project Files: 所有项目文件
- Project Production Files: 项目的代码文件
- Project Test Files: 项目的测试文件
- Open Files: 当前打开的文件
- Module 'Module Name': 主要的 *** 模块
- Current File: 当前文件

三、检查结果分析

左侧会提示你需要修改的地方, 点击某一条, 右侧则会出现关于该问题的详细描述。

▼ Android 66 errors 2903 warnings

▼ Lint 66 errors 2902 warnings

- ▶ Accessibility 288 warnings
- ▶ Correctness 66 errors 884 warnings
- ▼ Internationalization 1186 warnings
 - ▶ Bidirectional Text 797 warnings
 - ▶ Constant Locale 33 warnings
 - ▶ Hardcoded text 300 warnings
 - ▶ Overlapping items in RelativeLayout 12 warnings
 - ▶ TextView Internationalization 44 warnings
- ▶ Performance 452 warnings
- ▶ Security 19 warnings
- ▶ Usability 73 warnings
- ▶ onClick handler is missing in the related activity 1 warning

▶ General 2 errors 9 warnings 1 weak warning

▶ Groovy 4 warnings 8 weak warnings

300 problems: Suppress ▼

Hardcoded text inspection

Hardcoded text

Hardcoding text attributes directly in layout files is bad for several reasons:

* When creating configuration variations (for example for landscape or portrait) you have to repeat the actual text (and keep it up to date when making changes)

* The application cannot be translated to other languages by just adding new translations for existing string resources.

There are quickfixes to automatically extract this hardcoded string into a resource lookup.

Issue id: HardcodedText

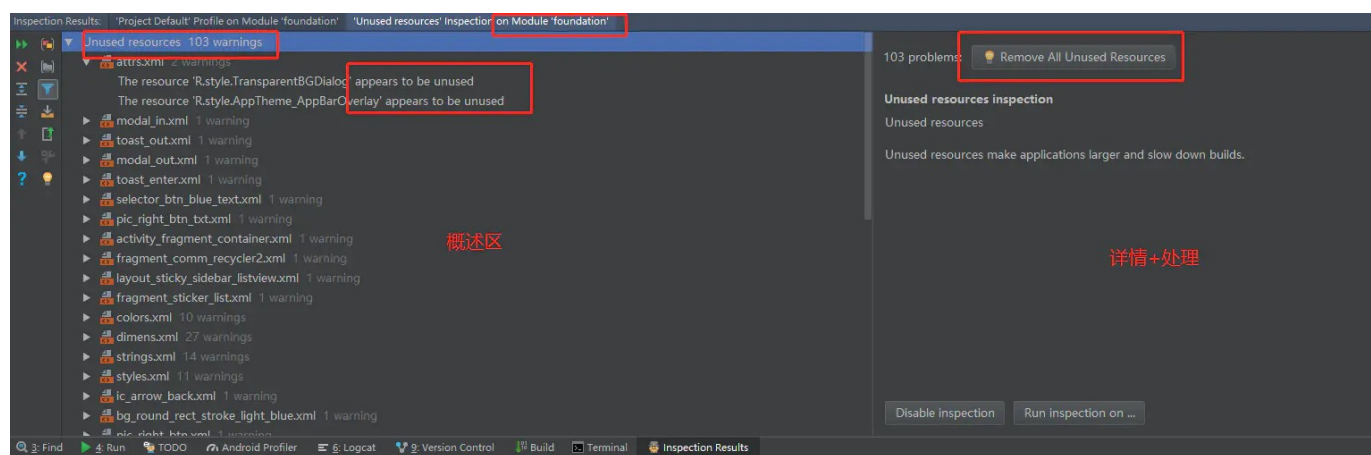
得到检查的结果后，就得开始对代码进行优化了，但是 Lint 报的某些警告的确是没必要的，这时我们可以选择忽略这些警告。因此其实在分析处理时，我们可以根据实际的需求进行忽略。

无用资源文件处理

随着不断地进行代码版本迭代，非常容易遗留一些无用的代码/源文件，这时可以通过直接指定检测无用资源，实现 Lint 清除。

当检测出的无用资源过多时，你会发现一个一个处理时多么的繁琐，这时可以注意到，下图右边有解决方法：Remove All Unused Resources，点击后，就没有了。

注：使用该方法后可能会造成某些资源误删的情况，因此删除之后不要忘记编译运行，利用 git 对其中误删资源进行 reset。



四、自定义Android Lint的检查提示

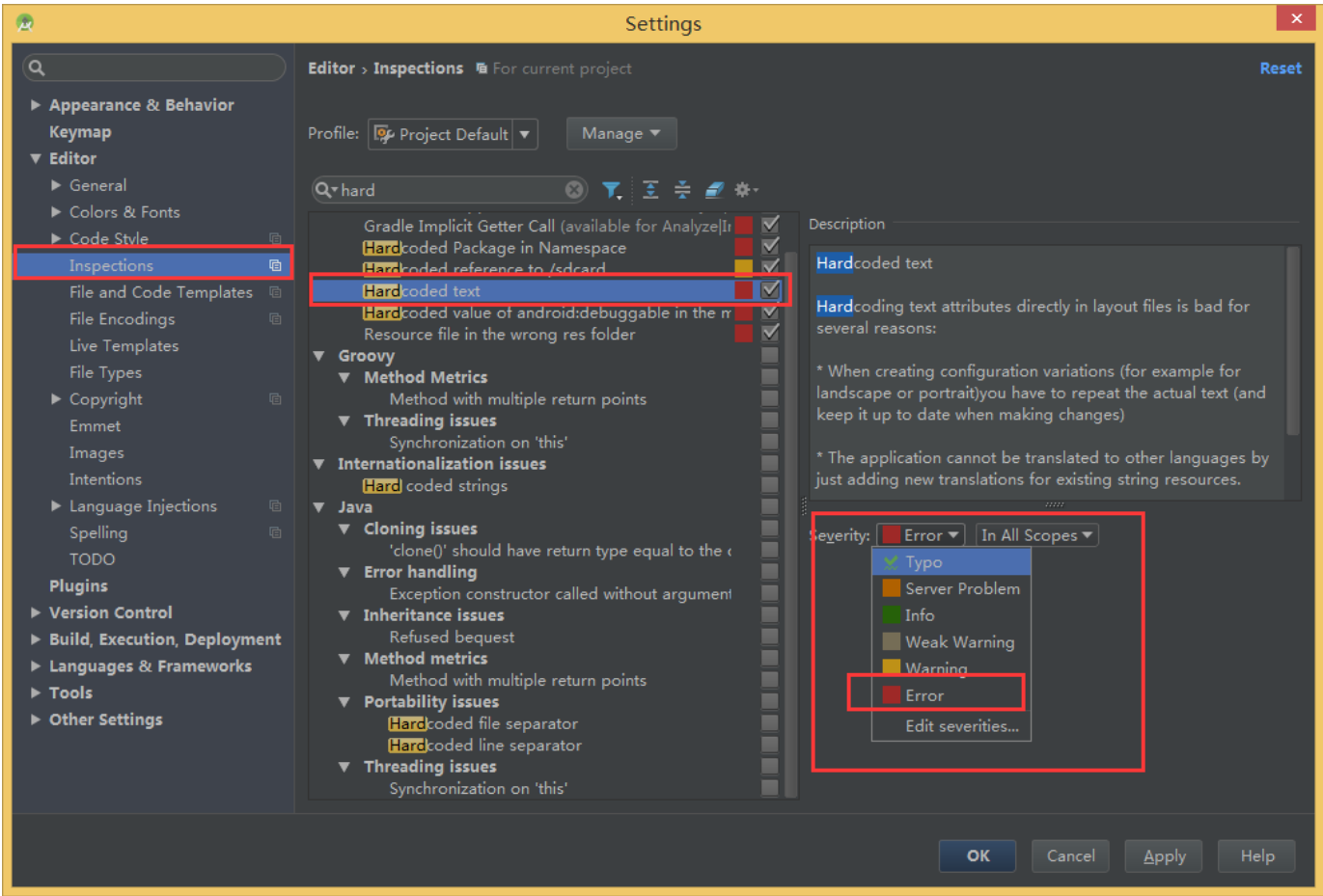
我们知道在xml文件中编写布局时候，假如我们在一个TextView中的text属性上直接写字符串，在textSize属性上写入的值用dp为单位，那么，此时AS中将会有建议提示，如：



可以看到这个提示太不明显了，所以我们可以通过更改对应的severity等级来更改提示的等

级，如：

默认hardcode的severity等级为warning，我们修改hardcode的severity等级为error，那么在存在硬编码时候将会以error等级提醒我们：



修改完成后，我们可以看到提示使用红色的波浪线标记了，更直观了，如：



当然这个只是起提示建议作用，并不会影响程序的正常运行。

五、Android-Lint所检查的常见问题

1) DuplicatedIds

Layout中id应该唯一

2) NewApi

代码中使用的某些API高于Manifest中的Min SDK

3) InconsistentArrays

字符串国际化中，同一名字的String-Array对应的item值不相同

4) Registered

Activity/Service/ContentProvider没有通过AndroidManifest注册

5) Deprecated

使用已经废弃的API

6) PxUsage

避免使用px，使用dp

2. Correctness:Messages

1) MissingTranslation

字符串国际化不完全

2) ExtraTranslation

国际化的字符串，在默认位置(defaultlocale)，没有定义

3. Security

1) SetJavaScriptEnabled

不确定你的程序中确实需要JavaScript就不要执行SetJavaScriptEnabled。

2)ExportedContentProvider/ExportedReceiver/ExportedService/ExportedActivity
ContentProvider/Receiver/Service/Activity的exported为true时，设置一个Permiss:

3) HardcodedDebugMode

不要在manifest中设置android:debuggable。

设置，编译的任何版本都要采用指定的debug模式。不设置，编译Eng版本采用debug模式；编

4. Performance

1) DrawAllocation

避免在绘制或者解析布局(draw/layout)时分配对象。eg:OnDraw()中实例化Paint对象。

2) ObsoleteLayoutParam

Layout中无用的参数。

3) UseCompoundDrawables

可优化的布局：如包含一个Imageview和一个TextView的线性布局，可被采用CompoundDrawa

4) UseSparseArrays

尽量用Android的SparseArray代替Hashmap。

5) DisableBaselineAlignment

如果LinearLayout被用于嵌套的layout空间计算，它的android:baselineAligned属性应该

6) FloatMath

使用FloatMath代替Math。

7) NestedWeights

避免嵌套weight，那将拖累执行效率。

8) UnusedResources/UnusedIds

未被使用的资源会是程序变大，并且编译速度降低。

9) Overdraw

如果为RootView指定一个背景Drawable，会先用Theme的背景绘制一遍，然后才用指定的背景。可以设置theme的background为null来避免。

10) UselessLeaf/UselessParent

View或view的父亲没有用。

5. Usability:Typography

1) TypographyDashes

特殊字符需用编码代替：“-”需要用“-”；“_”需要用“_”

2) TypographyEllipsis

特殊字符需用编码代替：“...”需要用“...”

3) TypographyOther

问题：“(c)”需要用“©”

6. Usability:Icons

1) IconNoDpi

Icon在nodpi和指定dpi的目录下都出现。

2) GifUsage

Image不要用GIF，最好用PNG，可以用JPG。

7. Usability

1) BackButton

Android中不要设计有Back的按钮，Android中一般有Back的硬按键。

2) ButtonCase

Button的“Ok”/“Cancel”显示大小写一定不要全大写或全小写。有标准的资源的字符串，不要

8. Accessibility

1) ContentDescription

ImageView和ImageButton应该提供contentDescription。

9. Internationalization

1) HardcodeText

硬编码的字符串应该在资源里定义。

2) EnforceUTF8

所有XML资源文件都应该以UTF-8编码。

