[如何利用JClassLib修改.class文件](http://blog.csdn.net/betterandroid/article/details/14520667)

標籤： [Android](http://www.csdn.net/tag/Android" \t "_blank)[jclasslib](http://www.csdn.net/tag/jclasslib)[反編譯](http://www.csdn.net/tag/%e5%8f%8d%e7%bc%96%e8%af%91)

2013-11-08 14:27 2945人閱讀 [評論](http://blog.csdn.net/betterandroid/article/details/14520667#comments)(2) [收藏](javascript:void(0);) [舉報](http://blog.csdn.net/betterandroid/article/details/14520667#report)

http://static.blog.csdn.net/images/category_icon.jpg 分類：

Android反編譯

版權聲明：本文為博主原創文章，未經博主允許不得轉載。

**最近在學習逆向分析和反編譯，無意之中瞭解到了JClassLib。JClassLib不但是一個字節碼閱讀器而且還包含一個類庫允許開發者讀取,修改,寫入Java Class文件與字節碼。其他的用途我就不說了，先看一下效果。**

第一步、準備下載工具，一個是jd-gui，閱讀jar包和.class源碼的工具，jclasslib就修改.class文件的包

[下載地址](http://download.csdn.net/detail/q59521038/6522461)

第二步、示範

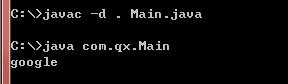
1、比如說這個就是原來的java文件，很簡單就是輸出 google 這個字符串

**[html]** [view plain](http://blog.csdn.net/betterandroid/article/details/14520667) [copy](http://blog.csdn.net/betterandroid/article/details/14520667)

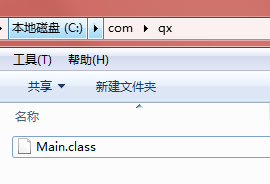
[print?](http://blog.csdn.net/betterandroid/article/details/14520667)

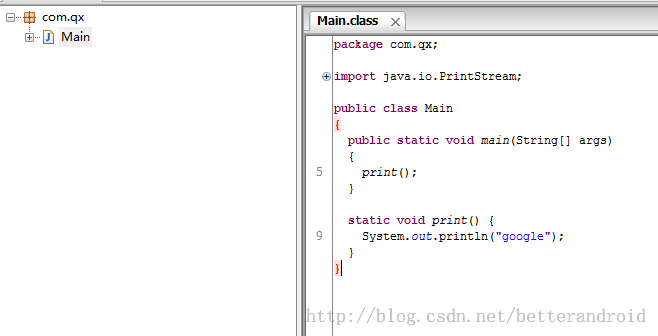
1. package com.qx;
3. public class Main {
4. public static void main(String[] args) {
5. print();
6. }
8. static void print() {
9. System.out.println("google");
10. }
11. }

2、編譯之後，控制台執行結果為 google，為什麼用 javac -d . Main.java 去編譯java文件，這個我就不解釋了，不懂的話去惡補一下

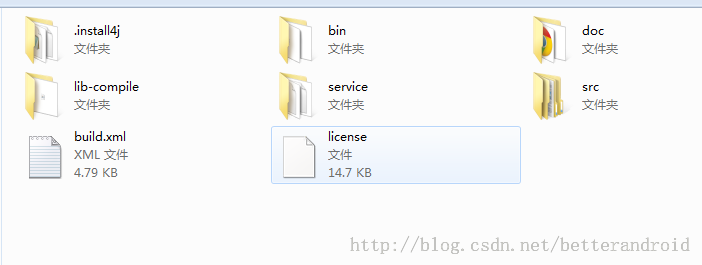


3、找到Main.class文件，並用 jd-gui打開，效果如下

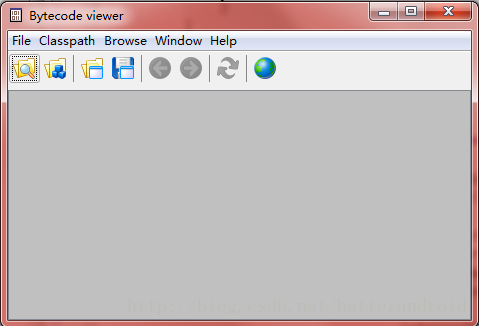
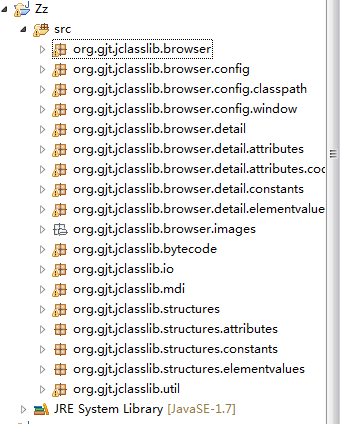




4、在這裡很明顯看到源碼，可惜不能修改呀，現在就準備修改工具，搞定它。解壓jclasslib，目錄如下

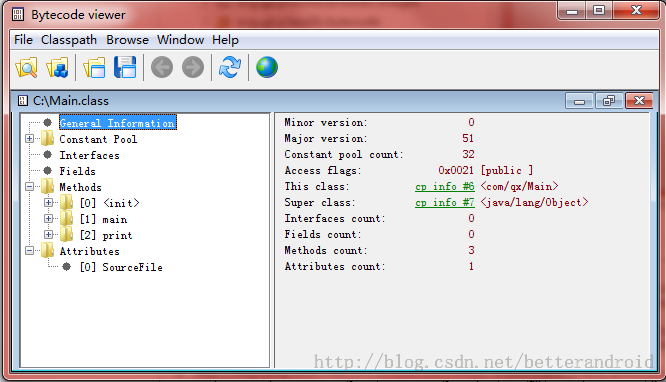


5、在eclipse裡面創建一個java工程，然後把src文件覆蓋到工程目錄下，並且執行BrowserApplication.java，得到一個工具叫做Bytecode viewer，如下圖

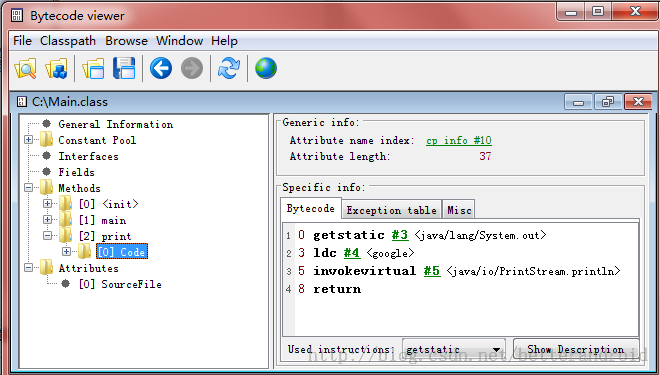


6、現在咱們可以利用工具找到我們修改的部分，我們就是要修改Main.class文件中google，修改成 baidu。

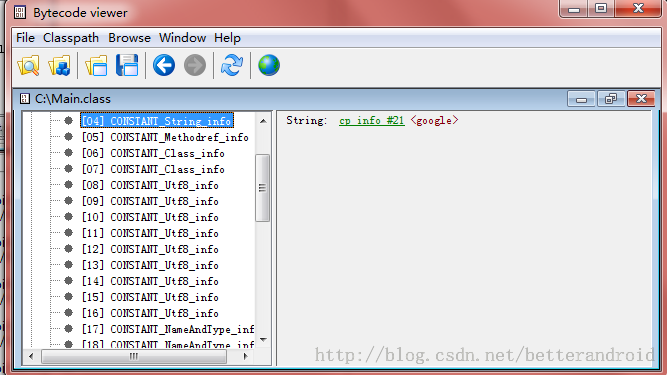
結構目錄很清楚，methods就是方法，print是打印google字符串的方法，main是主函數，在這裡我們找到print並打開



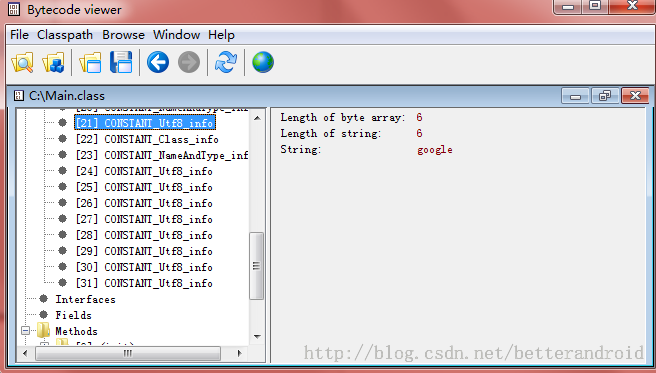
7、打開code，我們會看到第二行 lbc #4 <google>,這時我們點擊#4



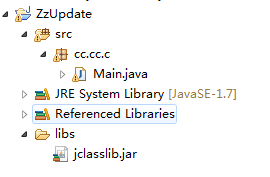
8、這個效果如下，通過這個，我們知道google字符串的地址在哪裡了，在#21， 這個時候 點擊 cp info #21



9、這時，我們看到的google字符是在[21]這個位置，並且CONSTANT\_Utf-8\_info，這些信息待會修改的時候 很有用。



10、現在我們可以用代碼去把 google 修改成 baidu。首先創建一個工程，把剛剛下載jclasslib文件夾下bin目錄下jclasslib.jar引入，如下圖



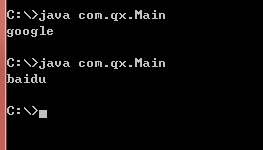
11、Main.java函數代碼如下

**[java]** [view plain](http://blog.csdn.net/betterandroid/article/details/14520667) [copy](http://blog.csdn.net/betterandroid/article/details/14520667)

[print?](http://blog.csdn.net/betterandroid/article/details/14520667)

1. **package** cc.cc.c;
3. **import** java.io.\*;
4. **import** org.gjt.jclasslib.io.ClassFileWriter;
5. **import** org.gjt.jclasslib.structures.CPInfo;
6. **import** org.gjt.jclasslib.structures.ClassFile;
7. **import** org.gjt.jclasslib.structures.constants.ConstantUtf8Info;
9. **public** **class** Main {
10. **public** **static** **void** main(String[] args) **throws** Exception {
12. String filePath = "C:\\Main.class";
13. FileInputStream fis = **new** FileInputStream(filePath);
14. DataInput di = **new** DataInputStream(fis);
15. ClassFile cf = **new** ClassFile();
16. cf.read(di);
17. CPInfo[] infos = cf.getConstantPool();
19. **int** count = infos.length;
20. **for** (**int** i = 0; i < count; i++) {
21. **if** (infos[i] != **null**) {
22. System.out.print(i);
23. System.out.print(" = ");
24. System.out.print(infos[i].getVerbose());
25. System.out.print(" = ");
26. System.out.println(infos[i].getTagVerbose());
27. **if** (i == 21) {//剛剛找到的是21位置
28. ConstantUtf8Info uInfo = (ConstantUtf8Info) infos[i]; //剛剛那裡是CONSTANT\_Utf-8\_info所以這裡要用這個
29. uInfo.setBytes("baidu".getBytes());
30. infos[i] = uInfo;
31. }
32. }
33. }
34. //這種方式也可以，一樣的
35. /\*      if(infos[count] != null) {
36. ConstantUtf8Info uInfo = (ConstantUtf8Info) infos[i]; //剛剛那裡是CONSTANT\_Utf-8\_info所以這裡要用這個
37. uInfo.setBytes("baidu".getBytes());
38. infos[count] = uInfo;
39. }\*/
41. cf.setConstantPool(infos);
42. fis.close();
43. File f = **new** File(filePath);
44. ClassFileWriter.writeToFile(f, cf);
45. }
46. }

12、不報錯，就代表成功了，然後再執行Main.class文件



這個時候，我們就把class文件中google修改成baidu了，怎麼用這個工具就看你怎麼玩了，有什麼問題請隨時留言。