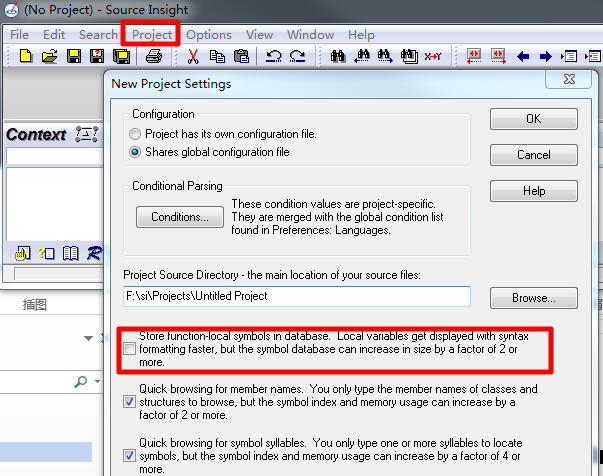
# Source Insight

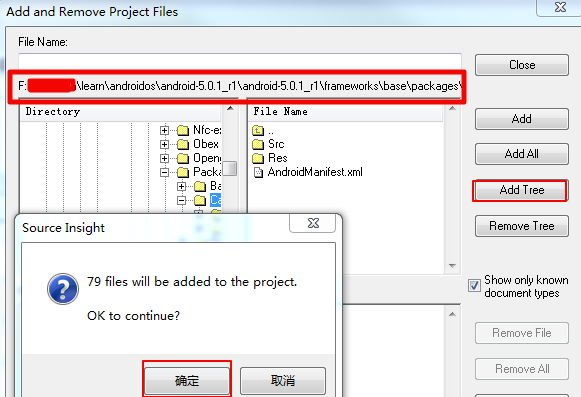
Source Insight的实质上是一个支持多种开发语言（java,c ,c++等等）的编辑器，只不过由于其查找、定位、彩色显示等功能的强大，而被我们当成源代码阅读工具使用，[下载和使用实例请点这里](https://jingyan.baidu.com/article/db55b609a8c6104ba30a2f88.html)。

## 新建工程

首先必须选择功 能菜单上的 “Project”选项的子菜单“New Project”新建一个项目，项目名称可以自由选定，当然也可以选择删除（Remove）一个项目。当删除一个项目的时候，并不删除原有的源代码文件, 只是将该软件生成的那些工程辅助文件删除



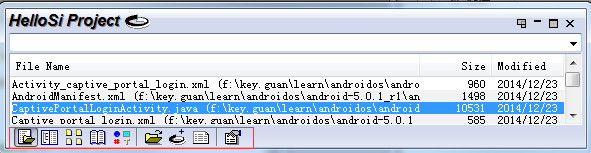
点击“OK”按钮，接受选择后，将会有一个新的对话框弹出，在这个对话框里，可以选择将要阅读的文件加入工程，一种方式是通过在File Name中输入要阅读源代码文件的名称，点击“Add”按钮将其加入，也可以通过其中“Add All”和“Add Tree”两个按钮可以将选中目录的所有文件加入到工程中，其中“Add All”选项会提示加入顶层文件和递归加入所有文件两种方式，而“**Add Tree**”相当于“Add All”选项的递归加入所有文件，可以根据需要使用。



添加完成之后，点击“close”退出添加。

## 工程视图

**默认可能没有打开Project View，需要从View->Project Views手动打开。**从左至右分别为：按字母顺序文件列表、显示文件夹、按照文件类型归类文件、全部文件的所有标记列表、按照标记类型归类标记、跳转到定义 处、显示标记信息、浏览工程标记、查找函数调用、工程属性。如下图



其中全部文件的所有标记列表选项可能要一段时间抽取标记，同步到数据库去，如果开始选择了建立 标记数据库，将会在今后节省同步时间，最有用的莫过于**浏览标记信息和查找函数调用（右二）**，前者可以通过“Jump”按钮在不同的地方查找同样的标志，还可以通过 “Reference”按钮结合后者进行全局的标记查找。

Reference 功能是Source Insight的特色之一，它可以在速度极快的在整个工程中找到所有的标记，并且在该行程序的前边加上红色箭头的小按钮链接上，

图 1‑1是一个 Reference搜索后的结果，它可以有两种模式，一种集中显示结果，**下图**显示的就是这种模式，在这种模式下，可以通过前边的红色（红2）箭头小按钮进入另外一 种模式，该标记的具体的所在处，也可以通过标记的具体所在处点击红色箭头小按钮进入警种模式，还可以通过工具条上的两个红色（红4）小箭头直接在第二种模式下前后 移动，察看相应信息。它的这个强大的功能使得阅读Linux源程序有如神助。但是要注意的是，当进行了第二次“Reference”时，它会提示你将结果 集附加在第一个结果集的后边还是取代第一个结果集。如果选择前者，不能对结果集根据前后两次搜索结果进行分类，然后在其子类里进行移动，只能在整个结果集 里移动；如果，选择后者，结果集将会被替换为第二次搜索的结果，略微有些不方便。

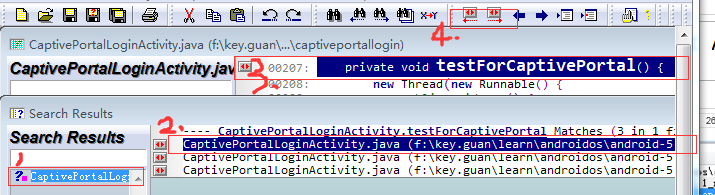


图 1‑1 Reference的搜索结果

右键菜单几乎包含了程序的所有功能，可以在编辑窗口为程序加上行号，还可以统计整个工程的程序行数

## 文件视图

点击一个文件就可以打开该文件，显示如图 1‑2所示，进入到右边的那个窗口分别可以以文件列表的方式，列出所有的文件，每个窗体下边有一排按钮，左边的窗口 （CaptivePortalLoginActivity.java）从左至右分别为：按字母顺序排列所有标记、按照文件中行数顺序排列标记、按照类型排列标记、浏览本地文件标记、标记窗口属性。如下图：

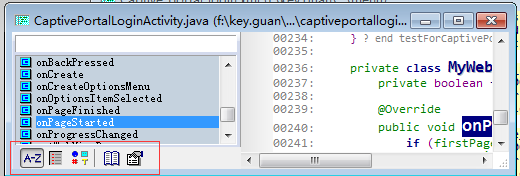
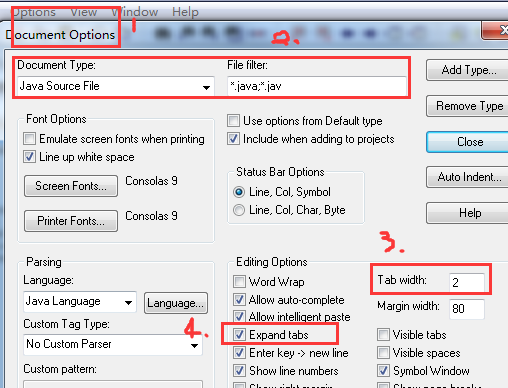


图 1‑2 文件视图

## 使用技巧

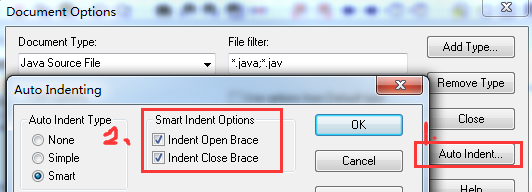
### 按tab键、等价于输入2个空格



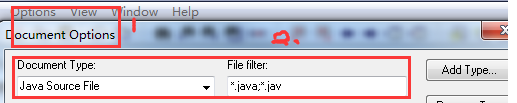
### 修改if语句缩进不正常

勾选Auto Indent之SMART的效果∶在C/java程序里, 如果遇到行末没有分号的语句,如IF, WHILE, SWITCH等, 写到该行末按回车，则新行自动相对上一行缩进两列. 输入"{"时，不会与下面的行对齐（这是勾上Indent Open Brace时的效果）

去掉Indent Open Brace和Indent Close Brace的效果∶在相对缩进行里, 如果输入"}", 则自动和上一行列对齐。操作如下：



### 向项目中只添加特定类型的文件（文件类型过滤器



### 自动补全

1. Options菜单->Preferences ->Typing卡->Auto Completion框，选择Use automatic symbol completion window（这里是SIS的全局设置）
2. Options菜单Document OptionsEditing Options框中，勾选Allow auto-complete（局部设置）

上面两项必须全部勾选，才能启用Auto Completion功能

### 修改快捷键

入口：Options菜单Key assignments，通过关键词查改相应快捷键。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 可以words |  | 备注 |
| 翻页设置 | Scroll Half Page Up |  |  |
| 函数定义偏移 | Function Up |  |  |
| 全选功能 | select | ctrl +a | ctrl +a默认是save all，因此需要手动修改 |
| 查找 |  | ctrl +F |  |
| 替换 |  | Ctrl+H | 勾选WholeFile来在整个当前文件内替换、两者都不勾选来从当前光标处替换至文件末尾；点右边的Files…按钮，可选择替换多个文件的内容。 |
| Smart Rename |  | Ctrl+' | 它可以智能地重命名全部项目文件中的一个标示符。SourceInsight的搜索索引（search index）使得搜索过程进行地非常快。而且，使用Smart Rename所做的替换会被记录在Search Results窗口中，每一条替换记录旁有一个SourceLink链接到替换发生地文件。  Confirm Each Replacement 每次替换询问。  Skip Comments 不重名注释部分。 |

### 正则表达式

|  |  |
| --- | --- |
| Character  Matches  ^ | (at the beginning)     beginning of line。如^Hello，匹配Hello在句首。 |
| . | any single character |
| [abc] | any single character that belongs to the set abc |
|  | [^abc]  any single character that does not belong to the set abc |
| \* | zero or more occurrences of the preceding character |
| + | a tab character |
| s | a space character |
| w | white space (a tab or a space character) |
| $ | the end of the line。如TRUE$，匹配TRUE在句尾。 |
| \ | 转义字符。如果在它后面有元字符，取消其特殊含义 |

可利用 "("和 ")"、把正则表达式分割成不同的组；模式中的每个组自左向右指定为 Group #n，n=1,2,…；组的概念在替换时很有用。  例如∶

abc(xyz)可匹配abcxyz，其中xyz被认为是group#1，

利用21来替换(abc)(xyz)，替换结果为xyzabc。  3.2 正则表达式在配置tc编译器中的应用∶

正则表达式格式与源代码文件路径相对应，这里我的tc安装目录为d:tc

更多请看参考链接。

### 自定义命令

分别利用上面SIS对输出信息的处理方式，自定义命令对集成编译器相当有用，可以捕捉编译器输出并创建SourceLink寻错；自定义命令对于文本过滤也相当有用，可选中待过滤区块、运行Sort自定义命令、粘贴回选定区块、即完成文本过滤。

请按下面步骤创建自定义命令∶  Options菜单Custom Command

点右边Add钮、填入新自定义命令名称，或下拉左边Commands、选择命令进行修改

Run框、填入待执行命令行，可含有特殊元字符，见后面的元字符表

Dir框、执行命令行时应处的目录，如不填，以源代码文件所在目录为命令执行目录

勾选Output框的Capture Output、输出被捕捉，如果勾选Paste Output，输出被粘贴

勾选Control Group框中的Save Files First、SIS会在运行命令前先检查文件是否保存

勾选Control Group框中的Pause When Done、SIS会在命令结束后暂停、方便检查

勾选Source Links in Output框中的Parse Source Links，?/p>

常用宏: 该宏文件实现一些编码程中能会到的功能, 如添加文件头、函数说明和宏定义等, 使用时能自动添加文件名、函数名和当前日期.

# ADB

Android Debug Bridge

启动/关闭adb服务:adb start-server/kill-server

多设备（模拟器）指定设备号: adb devices -l; adb –s <serialNumber>

## Pm

安装/卸载/清除apk：

adb install -r XX.apk；

adb unstall packagename

adb shell pm clear packagename 在root设备才生效！

降级安装down：adb install -r –d XX.apk

* adb shell pm list packages -f  
  查看所有已安装的应用的包名

## dumpsys

|  |  |  |
| --- | --- | --- |
|  |  |  |
| activity | 获取所用应用 | 获取所用应用:adb shell dumpsys activity  获取自己的应用:adb shell dumpsys activity | grep com.xxx.xxx.xx  获取处于栈顶的activity:adb shell dumpsys activity | grep mFocusedActivity |
| package |  |  |
| meminfo | 内存使用情况 |  |
| procstats | Memory Use Over Time |  |
| gfxinfo | Graphics State |  |

## 文件处理

adb pull <remote> <local>  
从手机复制文件出来

adb push <local> <remote>  
向手机发送文件eg. adb push foo.txt /sdcard/foo.txt

系统属性（/SYSTEM/BUILD.PROP）：adb shell getprop

日志重定向; adb logcat > bug1121.txt

## 调试

### 广播am

adb shell am broadcast -a android.intent.action.test.XX --ei ei3 1：发送传整数ei3 = 1d的XX广播

adb shell am start -n packageNanme/ActivityFullName：发送启动Activity的广播

## adb无法启动

原因：端口被占用，Unable to create Debug Bridge: Unable to start adb server: error: could not install \*smartsocket\* listener: cannot bind to 127.0.0.1:5037

解决办法：



参考：

[ANDROID开发常用的ADB命令整理kg](http://www.keyguan.com/blog/2017/02/11/android%E5%BC%80%E5%8F%91%E5%B8%B8%E7%94%A8%E7%9A%84adb%E5%91%BD%E4%BB%A4%E6%95%B4%E7%90%86/)

# 参考

[最经典的 Source\_Insight 完整使用教程](https://wenku.baidu.com/view/c32b120c6c85ec3a87c2c519.html/)

# QA

（1）Options菜单PreferencesTyping卡，勾掉下面两项∶

Typing tab indents line，regardless of selection，空行按tab无法前进

Typing tab replaces current selection，选定部分内容、再按tab时会清除所选