android adb常用指令

Android 调试桥(adb)是多种用途的工具，该工具可以帮助你你管理设备或模拟器 的状态。

可以通过下列几种方法加入adb:

- 在设备上运行shell命令
- 通过端口转发来管理模拟器或设备
- 从模拟器或设备上拷贝来或拷贝走文件

下面对adb进行了介绍并描述了常见的使用.

Contents

2011年11月 (2)
2011年10月 (3)
2011年9月 (2)
2011年8月 (10)
2011年7月 (3)
2011年6月 (3)
2011年5月 (11)
2011年4月 (15)
2011年3月 (18)
2010年12月 (1)
2010年11月 (4)
2010年10月 (1)
2010年9月 (5)
2010年8月 (4)

概要

Android 调试系统是一个面对客户服务系统，包括三个组成部分:

- 一个在你用于开发程序的电脑上运行的客户端。你可以通过shell端使用adb命令启动客户端。 其他Android工具比如说ADT插件和DDMS同样可以产生adb客户端.
- 在你用于发的机器上作为后台进程运行的服务器。该服务器负责管理客户端与运行于模拟器或设备上的adb守护程序(daemon)之间的通信。.
- 一个以后台进程的形式运行于模拟器或设备上的守护程序(daemon)。.

当你启动一个adb客户端，客户端首先确认是否已有一个adb服务进程在运行。如果没有，则启动服务进程。当服务器运行， adb服务器就会绑定本地的TCP端口5037并监听adb客户端发来的命令，—所有的adb客户端都是用端口 5037与adb服务器对话的.

接着服务器将所有运行中的模拟器或设备实例建立连接。它通过扫描所有5555到5585范围内的奇数端口来定位所有的模拟器或设备。一旦服务器找到 了adb守护程序，它将建立一个到该端口的连接。请注意任何模拟器或设备实例会取得两个连续的端口——一个偶数端口用来相应控制台的连接，和一个奇数端口 用来响应adb连接。比如说:

```
模拟器1，控制台：端口5554
模拟器1，Adb端口5555
控制台：端口 5556
Adb端口5557...
```

如上所示，模拟器实例通过5555端口连接adb，就如同使用5554端口连接控制台一样.

一旦服务器与所有模拟器实例建立连接，就可以使用adb命令控制和访问该实例。因为服务器管理模拟器/设备实例的连接，和控制处理从来自多个adb客户端来的命令，你可以通过任何客户端(或脚本)来控制任何模拟器或设备实例.

以下的部分描述通过命令使用adb和管理模拟器/设备的状态。要注意的是如果你用，装有ADT插件的Eclipse开发Android程序，你就不 需要通过命令行使用adb。ADT插件已经透明的把adb集成到Eclipse中了，当然，如果必要的话你也可以仍然直接使用adb，比如说调试.

发出**adb**命令

发出Android命令： 你可以在你的开发机上的命令行或脚本上发布Android命令，使用方法:

```
adb [-d|-e|-s <serialNumber>] <command>
```

当你发出一个命令，系统启用Android客户端。客户端并不与模拟器实例相关，所以如果双服务器/设备是运行中的，你需要用 -d 选项去为应被控制的命令确定目标实例。关于使用这个选项的更多信息，可以查看模拟器/设备实例术语控制命令 .

4. Android APK反编译(84754)
5. Android Launcher--简易Launcher开发(66262)

评论排行榜
1. Android APK反编译(21)
2. Android的线程使用来更新UI---Thread、Handler、Looper、TimerTask等(7)
3. Android widget 之RemoteView(7)
4. android adb常用指令(6)
5. Android的TextView使用Html来处理图片显示、字体样式、超链接等(6)

推荐排行榜
1. Android的线程使用来更新UI---Thread、Handler、Looper、TimerTask等(18)
2. android adb常用指令(13)
3. Android APK反编译(10)
4. JAVA正则表达式 Pattern和Matcher(8)
5. Android_launcher的源码详细分析(7)

查询模拟器/设备实例

在发布adb命令之前，有必要知道什么样的模拟器/设备实例与adb服务器是相连的。可以通过使用devices 命令来得到一系列相关联的模拟器/设备:

```
adb devices
```

•作为回应，adb为每个实例都制定了相应的状态信息:

- •序列号——由adb创建的一个字符串，这个字符串通过自己的控制端口<type>-<consolePort> 唯一地识别一个模拟器/设备实例。下面是一个序列号的例子： emulator-5554
- 实例的连接状态有三种状态:
  - offline — 此实例没有与adb相连接或者无法响应.
  - device — 此实例正与adb服务器连接。注意这个状态并不能百分之百地表示在运行和操作Android系统，因此这个实例是当系统正在运行的时候与adb连接的。然而，在系统启动之后，就是一个模拟器/设备状态的正常运行状态了.

每个实例的输出都有如下固定的格式:

```
[serialNumber] [state]
```

下面是一个展示devices 命令和输出的例子 :

```
$ adb devicesList of devices attached emulator-5554  deviceemulator-5556  deviceemulator-5558  device
```

如果当前没有模拟器/设备运行，adb则返回 no device .

给特定的模拟器/设备实例发送命令

如果有多个模拟器/设备实例在运行，在发布adb命令时需要指定一个目标实例。 这样做，请使用-s 选项的命令。在使用的-s 选项是

```
adb -s <serialNumber> <command>
```

如上所示，给一个命令指定了目标实例，这个目标实例使用由adb分配的序列号。你可以使用 devices 命令来获得运行着的模拟器/设备实例的序列号

示例如下:

```
adb -s emulator-5556 install helloWorld.apk
```

注意这点，如果没有指定一个目标模拟器/设备实例就执行 -s 这个命令的话，adb会产生一个错误.

安装软件

你可以使用adb从你的开发电脑上复制一个应用程序，并且将其安装在一个模拟器/设备实例。像这样做，使用install 命令。这个install 命令要求你必须指定你所要安装的.apk文件的路径:

```
adb install <path_to_apk>
```

为了获取更多的关于怎样创建一个可以安装在模拟器/设备实例上的.apk文件的信息，可参照Android Asset Packaging Tool (aapt).

要注意的是，如果你正在使用Eclipse IDE并且已经安装过ADT插件，那么就不需要直接使用adb（或者aapt）去安装模拟器/设备上的应用程序。否则，ADT插件代你全权处理应用程序的打包和安装.

转发端口

可以使用 forward 命令进行任意端口的转发——一个模拟器/设备实例的某一特定主机端口向另一不同端口的转发请求。下面演示了如何建立从主机端口6100到模拟器/设备端口7100的转发。

```
adb forward tcp:6100 tcp:7100
```

同样地，可以使用adb来建立命名为抽象的UNIX域套接口，上述过程如下所示:

```
adb forward tcp:6100 local:logd
```

从模拟器/设备中拷入或拷出文件

可以使用adbpull ,push 命令将文件复制到一个模拟器/设备实例的数据文件或是从数据文件中复制。install 命令只将一个.apk文件复制到一个特定的位置，与其不同的是，pull 和 push 命令可令你复制任意的目录和文件到一个模拟器/设备实例的任何位置。

从模拟器或者设备中复制文件或目录，使用(如下命):

```
adb pull <remote> <local>
```

将文件或目录复制到模拟器或者设备，使用（如下命令）

```
adb push <local> <remote>
```

在这些命令中，<local>和<remote>分别指通向自己的发展机（本地）和模拟器/设备实例（远程）上的目标文件/目录的路径

下面是一个例子：:

```
adb push foo.txt /sdcard/foo.txt
```

Adb命令列表

下列表格列出了adb支持的所有命令,并对它们的意义和使用方法做了说明.

Category	Command	Description	Comments
Options	-d	仅仅通过USB接口来管理abd.	如果不只是用USB接口来管理则返回错误.
	-e	仅仅通过模拟器实例来管理adb.	如果不是仅仅通过模拟器实例管理则返回错误.
	-s <serialNumber>	通过模拟器/设备的允许的命令号码来发送命令来管理adb (比如: "emulator-5556").	如果没有指定号码，则会报错.
General	devices	查看所有连接模拟器/设备的设施的清单.	查看 Querying for Emulator/Device Instances获取更多相关信息.
	help	查看adb所支持的所有命令。.	
	version	查看adb的版本序列号.	
Debug	logcat [<option>] [<filter-specs>]	将日志数据输出到屏幕上.	
	bugreport	查看bug的报告，如dumpsys ,dumpstate ,和logcat 信息。	
	jdwp	查看指定的设施的可用的JDWP信息.	可以用 forward jdwp:<pid> 端口映射信息来连接指定的JDWP进程.例如: adb forward tcp:8000 jdwp:472 jdb -attach localhost:8000
Data	install <path-to-apk>	安装Android为（可以模拟器/设施的数据文件.apk指定完整的路径）.	
	pull <remote> <local>	将指定的文件从模拟器/设施的拷贝到电脑上.	
	push <local> <remote>	将指定的文件从电脑上拷贝到模拟器/设备中.	
	forward <local> <remote>	用本地指定的端口通过socket方法远程连接模拟器/设施	端口需要描述下列信息: • tcp:<portnum> • local:<UNIX domain socket name> • dev:<character device name>

			• jdwp:<pid>
Ports and Networking	ppp <tty> [parm]...	通过USB运行ppp： • <tty> — the tty for PPP stream. For exampledev:/dev/omap_csmi_ttyl. • [parm]... &mdash zero or more PPP/PPPD options, such as defaultroute ,local , notty , etc. 需要提醒你的不能自动启动PDP连接.	
Scripting	get-serialno	查看adb实例的序列号.	查看 Querying for Emulator/Device Instances可以获得更多信息.
	get-state	查看模拟器/设施的当前状态.	
	wait-for-device	如果设备不联机就不让执行,--也就是实例状态是 device 时.	你可以提前把命令转载在adb的命令器中,在命令器中的命令在模拟器/设备连接之前是不会执行其它命令的. 示例如下: adb wait-for-device shell getprop 需要提醒的是这些命令在所有的系统启动启动起来之前是不会启动adb的 所以在所有的系统启动起来之前你也不能执行其它的命令. 比如：运用install 的时候就需要Android包，这些包只有系统完全启动。例如： adb wait-for-device install <app>.apk 上面的命令只有连接上了模拟器/设备连接上了adb服务才会被执行，而在Android系统完全启动前执行就会有错误发生.
Server	start-server	选择服务是否启动adb服务进程.	
	kill-server	终止adb服务进程.	
Shell	shell	通过远程shell命令来控制模拟器/设备实例.	查看 获取更多信息 for more information.
	shell [<shellCommand>]	连接模拟器/设施执行shell命令，执行完毕后退出远程shell端l.	

启动**shell**命令

Adb 提供了shell端，通过shell端你可以在模拟器或设备上运行各种命令。这些命令以2进制的形式保存在本地的模拟器或设备的文件系统中:

/system/bin/...

不管你是否完全进入到模拟器/设备的adb远程shell端，你都能 shell 命令来执行命令.

当没有完全进入到远程shell的时候，这样使用shell 命令来执行一条命令:

adb [-d|-e|-s {<serialNumber>}] shell <shellCommand>

在模拟器/设备中不用远程shell端时，这样使用shell 命 :

adb [-d|-e|-s {<serialNumber>}] shell

通过操作CTRL+D 或exit 就可以退出shell远程连接.

下面一些就将告诉你更多的关于shell命令的知识.

通过远程**shell**端运行**sqllite3**连接数据库

通过adb远程shell端，你可以通过Android软sqlite3 命令程序来管理数据库。sqlite3 工具包含了许多使用命令，比如：.dump 显示表的内容，.schema 可以显示出已经存在的表空间的SQL CREATE结果集。Sqlite3还允许你远程执行sql命令.

通过sqlite3 , 按照前几节的方法登陆模拟器的远程shell端，然后启动工具就可以使用sqlite3 命令。当sqlite3 启动以后，你还可以指定你想查看的数据库的完整路径。模拟器/设备实例会在文件夹中保存SQLite3数据库. /data/data/<package_name>/databases/ .

示例如下:

$ adb -s emulator-5554 shell# sqlite3 /data/data/com.example.google.rss.rssexample/databases/rssitems.dbSQLite version 3.3.12Enter ".help" for instructions.... enter commands, then quit...sqlite> .exit

当你启动sqlite3的时候，你就可以通过shell端发送 sqlite3 ,命令了。用exit 或 CTRL+D 退出adb远程shell端.

UI/软件 试验程序 **Monkey**

当Monkey程序在模拟器或设备运行的时候，如果用户出发了比如点击，触摸，手势或一些系统级别的事件的时候，它就会产生随机脉冲，所以可以用Monkey用随机重复的方法去负荷测试你开发的软件.

最简单的方法就是用用下面的命令来使用Monkey，这个命令将会启动你的软件并且触发500个事件.

$ adb shell monkey -v -p your.package.name 500

更多的关于命令Monkey的命令的信息，可以查看UI/Application Exerciser Monkey documentation page.

文档页面

其它的**shell**命令

下面的表格列出了一些adbshell命令，如果需要全部的命令和程序，可以启动模拟器实例并且用adb -help 命令 .

adb shell ls /system/bin

对大部门命令来说，help都是可用的.

Shell Command	Description	Comments
dumpsys	清除屏幕中的系统数据n.	Dalvik Debug Monitor Service (DDMS)工具提供了完整的调试、.
dumpstate	清除一个文件的状态.	
logcat [<option>]... [<filter-spec>]...	启动信息日志并且但因输出到屏幕上.	
dmesg	输出主要的调试信息到屏幕上.	
start	启动或重启一个模拟器/设备实例.	
stop	关闭一个模拟器/设备实例.	

启用**logcat**日志

Android日志系统提供了记录和查看系统调试信息的功能。日志都是从各种软件和一些系统的缓冲区中记录下来的，缓冲区可以通过 logcat 命令来查看和使用.

使用**logcat**命令

你可以用 logcat 命令来查看系统日志缓冲区的内容:

[adb] logcat [<option>] ... [<filter-spec>] ...

请查看Listing of logcat Command Options ，它对logcat命令有详细的描述 .

你也可以在你的电脑或运行在模拟器/设备上的远程adb shell端来使用logcat 命令，也可以在你的电脑上查看日志输出。

$ adb logcat

你也这样使用:

logcat

过滤日志输出

每一个输出的Android日志信息都有一个标签和它的优先级.

- 日志的标签是系统部件原始信息的一个简要的标志。（比如：“View”就是查看系统的标签）.
- 优先级有下列集中，是按照从低到高顺利排列的:
  - V — Verbose (lowest priority)
  - D — Debug
  - I — Info
  - W — Warning
  - E — Error
  - F — Fatal
  - S — Silent (highest priority, on which nothing is ever printed)

在运行logcat的时候在前两列的信息中你就可以看到 logcat 的标签列表和优先级别,它是这样标出的:<priority>/<tag> .

下面是一个logcat输出的例子,它的优先级就似乎I,标签就是ActivityManage:

```
I/ActivityManager(  585): Starting activity: Intent { action=android.intent.action...}
```

为了让日志输出能体现管理的级别,你还可以用过滤器来控制日志输出,过滤器可以帮助你描述系统的标签等级.

过滤器语句按照下面的格式描tag:priority ... , tag 表示是标签, priority 是表示标签的报告的最低等级. 从上面的tag的中可以得到日志的优先级. 你可以在过滤器中多次写tag:priority .

这些说明都只到空白结束。下面有一个列子，例子表示支持所有的日志信息，除了那些标签为”ActivityManager”和优先级为”Info”以上的和标签为” MyApp”和优先级为” Debug”以上的。 小等级,优先权报告为tag.

```
adb logcat ActivityManager:I MyApp:D *:S
```

上面表达式的最后的元素 *:S ,，是设置所有的标签为"silent"，所有日志只显示有"View" and "MyApp"的，用 *:S 的另一个用处是 能够确保日志输出的时候是按照过滤器的说明限制的，也让过滤器也作为一项输出到日志中.

下面的过滤语句指显示优先级为warning或更高的日志信息:

```
adb logcat *:W
```

如果你电脑上运行logcat ，相比在远程adbshell端，你还可以为环境变量ANDROID_LOG_TAGS :输入一个参数来设置默认的过滤

```
export ANDROID_LOG_TAGS="ActivityManager:I MyApp:D *:S"
```

需要注意的是ANDROID_LOG_TAGS 过滤器如果通过远程shell运行logcat 或用adb shell logcat 来运行模拟器/设备不能输出日志.

控制日志输出格式

日志信息包括了许多元数据域包括标签和优先级。可以修改日志的输出格式，所以可以显示出特定的元数据域。可以通过 -v 选项得到格式化输出日志的相关信息.

- brief — Display priority/tag and PID of originating process (the default format).
- process — Display PID only.
- tag — Display the priority/tag only.
- thread — Display process:thread and priority/tag only.
- raw — Display the raw log message, with no other metadata fields.
- time — Display the date, invocation time, priority/tag, and PID of the originating process.
- long — Display all metadata fields and separate messages with a blank lines.

当启动了logcat ，你可以通过-v 选项来指定输出格式:

```
[adb] logcat [-v <format>]
```

下面是用 thread 来产生的日志格式:

```
adb logcat -v thread
```

需要注意的是你只能-v 选项来规定输出格式 option.

查看可用日志缓冲区

Android日志系统有循环缓冲区，并不是所有的日志系统都有默认循环缓冲区。为了得到日志信息，你需要通过-b 选项来启动logcat 。如果要使用循环缓冲区，你需要查看剩余的循环缓冲期:

- radio — 查看缓冲区的相关的信息.
- events — 查看和事件相关的的缓冲区.
- main — 查看主要的日志缓冲区

-b 选项使用方法:

```
[adb] logcat [-b <buffer>]
```

下面的例子表示怎么查看日志缓冲区包含radio 和 telephony信息:

```
adb logcat -b radio
```

查看**stdout** 和**stderr**

在默认状态下，Android系统有stdout 和 stderr (System.out 和System.err)输出到/dev/null ，在运行Dalvik VM的进程中，有一个系统可以备份日志文件。在这种情况下，系统会用stdout 和stderr 和优先级 I.来记录日志信息

通过这种方法指定输出的路径，停止运行的模拟器/设备，然后通过用 setprop 命令远程输入日志

$ adb shell stop$ adb shell setprop log.redirect-stdio true$ adb shell start

系统直到你关闭模拟器/设备前设置会一直保留，可以通过添加/data/local.prop 可以使用模拟器/设备上的默认设置

Logcat命令列表

Option	Description
-b <buffer>	加载一个可使用的日志缓冲区供查看，比如event 和radio . 默认值是main 。具体查看Viewing Alternative Log Buffers.
-c	清楚屏幕上的日志.
-d	输出日志到屏幕上.
-f <filename>	指定输出日志信息的<filename>，默认是stdout .
-g	输出指定的日志缓冲区，输出后退出.
-n <count>	设置日志的最大数目<count>.，默认值是4，需要和 -r 选项一起使用。
-r <kbytes>	每<kbytes> 时输出日志，默认值为16，需要和-f 选项一起使用.
-s	设置默认的过滤级别为silent.
-v <format>	设置日志输入格式，默认的是brief 格式，要知道更多的支持的格式，参看Controlling Log Output Format .

Stopping the adb Server

在某些情况下，你可能需要终止Android 调试系统的运行,然后再重新启动它。 例如,如果Android 调试系统不响应命令，你可以先终止服务器然后再重启，这样就可能解决这个问题.

用kill-server 可以终止adb server。你可以用adb发出的任何命令来重新启动服务器.

转载自：http://www.javaeye.com/topic/260042

分类: Android 资料

好文要顶 关注我 收藏该文

山岭巨人
关注 - 0
粉丝 - 318
+加关注

13 0

« 上一篇：Android SDCard操作(文件读写,容量计算)
» 下一篇：OpenGL教程（收集的一些好教程）

posted on 2010-09-19 10:22 山岭巨人 阅读(131074) 评论(6) 编辑 收藏

评论

#1楼 2011-12-05 13:58 john23.net

顶个

支持(0) 反对(0)

#2楼 2012-09-29 09:31 scwsmile

thx!

支持(0) 反对(0)

#3楼 2012-11-21 14:45 jason-_-shi

tks very much

支持(0) 反对(0)

#4楼 2013-06-10 14:20 mikola

good. 非常好.

支持(0) 反对(0)

#5楼 2015-06-04 09:49 snow__wolf

不错，很详细！

支持(0) 反对(0)

#6楼 2015-07-22 14:25 贺满

nice

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册，访问网站首页。

最新**IT**新闻：
· 宣布一年跳票半年，《王者荣耀》Switch版确认9月上线

· 虚拟现实行业的“工伤”：戴头盔太久，天天头晕想吐
· 美图2018上半年营收20.52亿 全面转型社交化
· HTC Vive/Vive Pro无线升级套装公布价格：9月5日发售
· 美学者揭露谷歌“疯狂”采集隐私：安卓手机每天“上缴”12MB
» 更多新闻...

最新知识库文章：
· 一个故事看懂“区块链”
· 被踢出去的用户
· 成为一个有目标的学习者
· 历史转折中的“杭派工程师”
· 如何提高代码质量？
» 更多知识库文章...