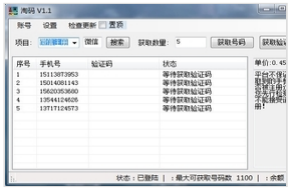
Android adb bugreport工具分析和使用

2016-07-20 09:07:44 0个评论 来源：Android Fans 收藏 我要投稿


短信验证码平台


沉香价格


手表品牌排行榜


什么食物降血糖

现在买什么手机好 灵芝一斤多少钱 人工智能培训 虫草价格表 冬虫夏草价格 无月租手机卡
明天股市走势预测 铁皮石斛价格 三七能降血糖吗 十大电吹风 透明手机价格 海参价格

bugreport是什么，怎么用？

Android系统想要成为一个功能完备，生态繁荣的操作系统，那就必须提供完整的应用开发环境。而在应用开发中，app程序的调试分析是日常生产中进程会进行的工作。Android为了方便开发人员分析整个系统平台和某个app在运行一段时间之内的所有信息，专门开发了bugreport工具。这个工具使用起来十分简单，只要在终端执行（linux或者win）：

```
1 <code class="hljs avrasm">adb bugreport > bugreport.txt</code>
```


海参价格

文章 推荐

· win7激活工具
· win10激活工具
· win7激活工具旗舰版
· office2010激活密钥
· windows7激活密钥
· office2010激活工具
· 小马激活工具
· win10激活工具

即可生成bugreport文件。但是有一个问题是，这个生成的文件有的时候异常庞大，能够达到15M+,想一想对于一个txt文本格式的文件内容长度达到了15M+是一个什么概念，如果使用文本工具打开查看将是一个噩梦。因此google针对android 5.0（api 21）以上的系统开发了一个叫做battery historian的分析工具，这个工具就是用来解析这个txt文本文件，然后使用web图形的形式展现出来，这样出来的效果更加人性化，更加可读。它的基本界面像下面这个样子：

查看标识获取更多信息
图片相关推广
美易财咖 第2季 争霸赛
市场有风险 投资需谨慎

目前google已经将bettery histZ喎"/kf/ware/vc/" target="_blank" class="keylink">vcmlhbr+q1LTBy6Osv6rUtM/uxL+1xLXY1rejujxiciAvPg0KPGEgaHJlZj0="https://github.co historian">https://github.com/google/battery-historian

google写了一个比较详细的说明文档，大家可以自行查阅一下。这个工具可以查看以下信息：

```
<code class="hljs avrasm"><code class="hljs applescript">Brightness
CPU running
Charging on
Charging status
Health
JobScheduler
Kernel only uptime
Level
```

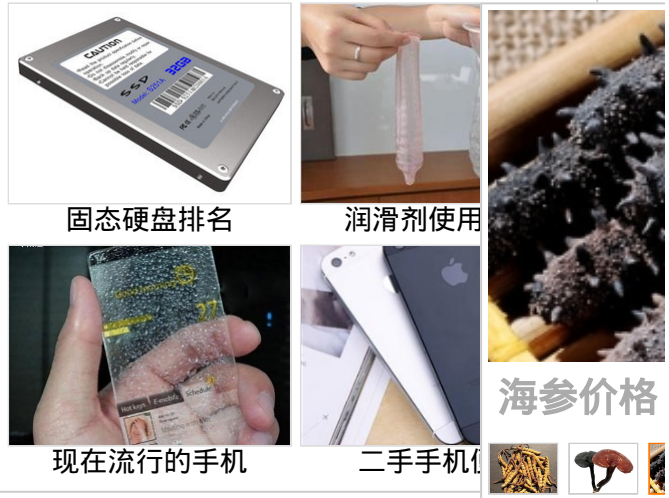
点击排行

- Android Studio安装配置详细步骤（图
- Android Studio中如何引用图片资源
- RecyclerView完全解析,让你从此爱上它
- ［Android Studio 权威教程］断点调
- Android之开发常用颜色
- Android M新控件之AppBarLayout，Nav
- 最新2017（Android）安卓面试题级答案
- Android Studio打包全攻略---从入门到

```
Package active
Partial wakelock
Phone scanning
Phone state
Plug
Plugged
Screen
Temperature
Top app
Voltage
Wifi on
Wifi running
Wifi supplicant</code></code>
```

数据还是比较详细的。

当然，同样的bugreport数据也可以有不同的解析和阅读方式，你如果不太喜欢google的battery historian的话，你还有别的选择，那就是选择Sony开源的ChkBugReport，这个工具提供了不同于battery historian的视角去解读bugreport文件，见面简单明了：



这个项目的文档：

http://developer.sonymobile.com/2012/01/25/new-bugreport-analysis-tool-released-as-open-source/

开源地址首页：

https://github.com/sonyxperiadev/ChkBugReport

这里说明一下，笔者使用过ChkBugReport这个工具，感觉很不错，最好结合google的battery historian；另外ChkBugReport这个工具还有一点bug，不过不影响使用。

bugreport的原理是什么？


海参价格


下面我们简要分析一下adb bugreport运行的原理。我们知道，使用bugreport只要执行adb bugreport命令就可以了，因此我们的分析肯定是从adbd这个daemon进程开始，我们查看这个进程的代码的时候发现这里处理了bugreport选项：

adb_commandline@system/core/adb/commandline.cpp

```
else if (!strcmp(argv[0], "bugreport")) {
    if (argc != 1) return usage();
    return send_shell_command(ttype, serial, "shell:bugreport");
}
```

查看标识获取更多信息
图片相关推荐

我们可以清楚地看到，这里判断如果附带的参数是bugreport的话，send_shell_command函数处理，这个函数的代码比较简单，我们就不分析了，这个函数的功能就是使用shell执行参数中的命令，因此我们这里相当于执行了bugreport命令。

在android设备中，bugreport命令存在于system/bin/目录下，这是一个可执行文件，所以我们要查看这个可执行文件实现的地方，它的实现代码在/frameworks/native/cmds/bugreport/目录下：

Name		Date	Size
..		13-Oct-2015	4 KiB
Android.mk	H A D	13-Oct-2015	198
bugreport.cpp	H A D	13-Oct-2015	2.9 KiB

我们看到，bugreport的实现是比较简单的，只有一个Android.mk和一个cpp实现代码，我们先看一下Android.mk文件：

```
LOCAL_PATH:= $(call my-dir)
include $(CLEAR_VARS)

LOCAL_SRC_FILES:= bugreport.cpp

LOCAL_MODULE:= bugreport

LOCAL_CFLAGS := -Wall

LOCAL_SHARED_LIBRARIES := libcutils

include $(BUILD_EXECUTABLE)
```


海参价格

这里我们看到该目录下的代码会被编译成一个名字叫做bugreport的可执行文件，这就是我们想要找的。现在我们看一下bugreport.cpp文件的实现，这个文件中代码比较简单，只有一个main函数：

```
<code class="hljs avrasm"><code class="hljs applescript"><code clas?
// dumpstate, then connect to the dumpstate local client to read the
// output. All of the dumpstate output is written to stdout, includir
// any errors encountered while reading/writing the output.
int main() {
  // Start the dumpstate service.
  property_set("ctl.start", "dumpstate");

  // Socket will not be available until service starts.
  int s;
  for (int i = 0; i < 20; i++) {
    s = socket_local_client("dumpstate", ANDROID_SOCKET_NAMESPACE_RES
                            SOCK_STREAM);
    if (s >= 0)
      break;
    // Try again in 1 second.
    sleep(1);
  }

  if (s == -1) {
    printf("Failed to connect to dumpstate service: %s\n", strerror(e
    return 1;
  }

  // Set a timeout so that if nothing is read in 3 minutes, we'll stc
  // reading and quit. No timeout in dumpstate is longer than 60 secc
  // so this gives lots of leeway in case of unforeseen time outs.
  struct timeval tv;
  tv.tv_sec = 3 * 60;
  tv.tv_usec = 0;
  if (setsockopt(s, SOL_SOCKET, SO_RCVTIMEO, &tv, sizeof(tv)) == -1)
    printf("WARNING: Cannot set socket timeout: %s\n", strerror(errnc
  }

  while (1) {
    char buffer[65536];
    ssize_t bytes_read = TEMP_FAILURE_RETRY(read(s, buffer, sizeof(bu
    if (bytes_read == 0) {
      break;
    } else if (bytes_read == -1) {
      // EAGAIN really means time out, so change the errno.
```


海参价格

```
      if (errno == EAGAIN) {
        errno = ETIMEDOUT;
      }
      printf("\nBugreport read terminated abnormally (%s).\n", strerr
      break;
    }

    ssize_t bytes_to_send = bytes_read;
    ssize_t bytes_written;
    do {
      bytes_written = TEMP_FAILURE_RETRY(write(STDOUT_FILENO,
                                                buffer + bytes_read -
                                                bytes_to_send));
      if (bytes_written == -1) {
        printf("Failed to write data to stdout: read %zd, trying to s
               bytes_read, bytes_to_send, strerror(errno));
        return 1;
      }
      bytes_to_send -= bytes_written;
    } while (bytes_written != 0 && bytes_to_send > 0);
  }

  close(s);
  return 0;
}</code></code></code>
```

这里的代码非常简单，主要的逻辑就是：

1.启动dumpstate service

2. 和dumpstate service建立socket链接

3. 从socket中读取数据，并且答应到stdout中

4. 读取完成之后关闭socket，然后退出

因此，我们分析的重点需要转移到dumpstate中了。这里说明一下，前面启动dumpstate service的方法是使用系统属性来实现，这个属性的改变消息会被init进程收到，然后init进程会启动dumpstate这个服务。

dumpstate其实也是一个可执行文件，也存在于system/bin目录下。现在我们明白了，其实bugreport就是dumpstate，只是bugreport将dumpstate包装了一下而已。


海参价格

现在我们需要分析一下dumpstate的实现，它的实现代码在：frameworks/native/cmds/dumpstate目录下，我们看下这个目录下的代码结构：

Name		Date	Size
..		13-Oct-2015	4 KiB
Android.mk	H A D	13-Oct-2015	510
dumpstate.c	H A D	13-Oct-2015	26.2 KiB
dumpstate.h	H A D	13-Oct-2015	2.8 KiB
libdumpstate_default.c	H A D	13-Oct-2015	677
utils.c	H A D	13-Oct-2015	24.1 KiB

这里的代码也是十分简单，只要少数的几个实现文件，其中main函数在dumpstate.c文件中，这个main函数我们这里不详细分析了，总结下它的主要工作：

1. 根据启动参数，初始化相关资源
2. 如果启动参数中带有-s的话（init启动会加上这个参数），就表示使用socket，那么就启动socket，并且在这个socket中等待链接。
3. 如果client端（也就是bugreport进程）链接成功，那就初始化所要用到的内存，并且设置优先级为较高优先级，防止被OOM干掉。
4. 然后使用vibrator震动一下（如果设备有这个硬件的话），提示用户开始截取log了
5. 调用dumpstate函数开始真正的dump工作
6. dump完成之后再次调用vibrator震动3次，提示用户dump完成。

现在我们看下dumpstate函数的实现：

```
<code class="hljs avrasm"><code class="hljs applescript"><code cla?
static void dumpstate() {
    unsigned long timeout;
    time_t now = time(NULL);
    char build[PROPERTY_VALUE_MAX], fingerprint[PROPERTY_VALUE_MAX];
    char radio[PROPERTY_VALUE_MAX], bootloader[PROPERTY_VALUE_MAX];
    char network[PROPERTY_VALUE_MAX], date[80];
    char build_type[PROPERTY_VALUE_MAX];

    property_get("ro.build.display.id", build, "(unknown)");
    property_get("ro.build.fingerprint", fingerprint, "(unknown)");
    property_get("ro.build.type", build_type, "(unknown)");
```


海参价格


```
    property_get("ro.baseband", radio, "(unknown)");
    property_get("ro.bootloader", bootloader, "(unknown)");
    property_get("gsm.operator.alpha", network, "(unknown)");
    strftime(date, sizeof(date), "%Y-%m-%d %H:%M:%S", localtime(&now

    printf("============================================================
    printf("== dumpstate: %s\n", date);
    printf("============================================================

    printf("\n");
    printf("Build: %s\n", build);
    printf("Build fingerprint: '%s'\n", fingerprint); /* format is :
    printf("Bootloader: %s\n", bootloader);
    printf("Radio: %s\n", radio);
    printf("Network: %s\n", network);

    printf("Kernel: ");
    dump_file(NULL, "/proc/version");
    printf("Command line: %s\n", strtok(cmdline_buf, "\n"));
    printf("\n");

    dump_dev_files("TRUSTY VERSION", "/sys/bus/platform/drivers/trus
    run_command("UPTIME", 10, "uptime", NULL);
    dump_files("UPTIME MMC PERF", mmcblk0, skip_not_stat, dump_stat_
    dump_file("MEMORY INFO", "/proc/meminfo");
    run_command("CPU INFO", 10, "top", "-n", "1", "-d", "1", "-m", '
    run_command("PROCRANK", 20, "procrank", NULL);
    dump_file("VIRTUAL MEMORY STATS", "/proc/vmstat");
    dump_file("VMALLOC INFO", "/proc/vmallocinfo");
    dump_file("SLAB INFO", "/proc/slabinfo");
    dump_file("ZONEINFO", "/proc/zoneinfo");
    dump_file("PAGETYPEINFO", "/proc/pagetypeinfo");
    dump_file("BUDDYINFO", "/proc/buddyinfo");
    dump_file("FRAGMENTATION INFO", "/d/extfrag/unusable_index");

    dump_file("KERNEL WAKELOCKS", "/proc/wakelocks");
    dump_file("KERNEL WAKE SOURCES", "/d/wakeup_sources");
    dump_file("KERNEL CPUFREQ", "/sys/devices/system/cpu/cpu0/cpufre
    dump_file("KERNEL SYNC", "/d/sync");

    run_command("PROCESSES", 10, "ps", "-P", NULL);
    run_command("PROCESSES AND THREADS", 10, "ps", "-t", "-p", "-P",
    run_command("PROCESSES (SELINUX LABELS)", 10, "ps", "-Z", NULL);
    run_command("LIBRANK", 10, "librank", NULL);

    do_dmesg();

```


海参价格


```
    run_command("LIST OF OPEN FILES", 10, SU_PATH, "root", "lsof", N
    for_each_pid(do_showmap, "SMAPS OF ALL PROCESSES");
    for_each_tid(show_wchan, "BLOCKED PROCESS WAIT-CHANNELS");

    if (screenshot_path[0]) {
        ALOGI("taking screenshot\n");
        run_command(NULL, 10, "/system/bin/screencap", "-p", screens
        ALOGI("wrote screenshot: %s\n", screenshot_path);
    }

    // dump_file("EVENT LOG TAGS", "/etc/event-log-tags");
    // calculate timeout
    timeout = logcat_timeout("main") + logcat_timeout("system") + lo
    if (timeout < 20000) {
        timeout = 20000;
    }
    run_command("SYSTEM LOG", timeout / 1000, "logcat", "-v", "threa
    timeout = logcat_timeout("events");
    if (timeout < 20000) {
        timeout = 20000;
    }
    run_command("EVENT LOG", timeout / 1000, "logcat", "-b", "events
    timeout = logcat_timeout("radio");
    if (timeout < 20000) {
        timeout = 20000;
    }
    run_command("RADIO LOG", timeout / 1000, "logcat", "-b", "radio'

    run_command("LOG STATISTICS", 10, "logcat", "-b", "all", "-S", N

    /* show the traces we collected in main(), if that was done */
    if (dump_traces_path != NULL) {
        dump_file("VM TRACES JUST NOW", dump_traces_path);
    }

    /* only show ANR traces if they're less than 15 minutes old */
    struct stat st;
    char anr_traces_path[PATH_MAX];
    property_get("dalvik.vm.stack-trace-file", anr_traces_path, "");
    if (!anr_traces_path[0]) {
        printf("*** NO VM TRACES FILE DEFINED (dalvik.vm.stack-trace
    } else {
      int fd = TEMP_FAILURE_RETRY(open(anr_traces_path,
                                       O_RDONLY | O_CLOEXEC | O_NOFO
      if (fd < 0) {
          printf("*** NO ANR VM TRACES FILE (%s): %s\n\n", anr_trace
      } else {
```


海参价格

```
            dump_file_from_fd("VM TRACES AT LAST ANR", anr_traces_pat
        }
    }

    /* slow traces for slow operations */
    if (anr_traces_path[0] != 0) {
        int tail = strlen(anr_traces_path)-1;
        while (tail > 0 && anr_traces_path[tail] != '/') {
            tail--;
        }
        int i = 0;
        while (1) {
            sprintf(anr_traces_path+tail+1, "slow%02d.txt", i);
            if (stat(anr_traces_path, &st)) {
                // No traces file at this index, done with the file
                break;
            }
            dump_file("VM TRACES WHEN SLOW", anr_traces_path);
            i++;
        }
    }

    int dumped = 0;
    for (size_t i = 0; i < NUM_TOMBSTONES; i++) {
        if (tombstone_data[i].fd != -1) {
            dumped = 1;
            dump_file_from_fd("TOMBSTONE", tombstone_data[i].name, 
            tombstone_data[i].fd = -1;
        }
    }
    if (!dumped) {
        printf("*** NO TOMBSTONES to dump in %s\n\n", TOMBSTONE_DIR
    }

    dump_file("NETWORK DEV INFO", "/proc/net/dev");
    dump_file("QTAGUID NETWORK INTERFACES INFO", "/proc/net/xt_qtagu
    dump_file("QTAGUID NETWORK INTERFACES INFO (xt)", "/proc/net/xt_
    dump_file("QTAGUID CTRL INFO", "/proc/net/xt_qtaguid/ctrl");
    dump_file("QTAGUID STATS INFO", "/proc/net/xt_qtaguid/stats");

    if (!stat(PSTORE_LAST_KMSG, &st)) {
        /* Also TODO: Make console-ramoops CAP_SYSLOG protected. */
        dump_file("LAST KMSG", PSTORE_LAST_KMSG);
    } else {
        /* TODO: Make last_kmsg CAP_SYSLOG protected. b/5555691 */
        dump_file("LAST KMSG", "/proc/last_kmsg");
    }
```


海参价格

```
    /* kernels must set CONFIG_PSTORE_PMSG, slice up pstore with dev
    run_command("LAST LOGCAT", 10, "logcat", "-L", "-v", "threadtime
                                              "-b", "all", "-d", "*:v

    /* The following have a tendency to get wedged when wifi drivers

    run_command("NETWORK INTERFACES", 10, "ip", "link", NULL);

    run_command("IPv4 ADDRESSES", 10, "ip", "-4", "addr", "show", NU
    run_command("IPv6 ADDRESSES", 10, "ip", "-6", "addr", "show", NU

    run_command("IP RULES", 10, "ip", "rule", "show", NULL);
    run_command("IP RULES v6", 10, "ip", "-6", "rule", "show", NULL)

    dump_route_tables();

    run_command("ARP CACHE", 10, "ip", "-4", "neigh", "show", NULL);
    run_command("IPv6 ND CACHE", 10, "ip", "-6", "neigh", "show", NU

    run_command("IPTABLES", 10, SU_PATH, "root", "iptables", "-L", '
    run_command("IP6TABLES", 10, SU_PATH, "root", "ip6tables", "-L",
    run_command("IPTABLE NAT", 10, SU_PATH, "root", "iptables", "-t'
    /* no ip6 nat */
    run_command("IPTABLE RAW", 10, SU_PATH, "root", "iptables", "-t'
    run_command("IP6TABLE RAW", 10, SU_PATH, "root", "ip6tables", "-

    run_command("WIFI NETWORKS", 20,
            SU_PATH, "root", "wpa_cli", "IFNAME=wlan0", "list_networ

#ifdef FWDUMP_bcmdhd
    run_command("ND OFFLOAD TABLE", 5,
            SU_PATH, "root", "wlutil", "nd_hostip", NULL);

    run_command("DUMP WIFI INTERNAL COUNTERS (1)", 20,
            SU_PATH, "root", "wlutil", "counters", NULL);

    run_command("ND OFFLOAD STATUS (1)", 5,
            SU_PATH, "root", "wlutil", "nd_status", NULL);

#endif
    dump_file("INTERRUPTS (1)", "/proc/interrupts");

    run_command("NETWORK DIAGNOSTICS", 10, "dumpsys", "connectivity'

#ifdef FWDUMP_bcmdhd
    run_command("DUMP WIFI STATUS", 20,
```


海参价格

```
            SU_PATH, "root", "dhdutil", "-i", "wlan0", "dump", NULL
    run_command("DUMP WIFI INTERNAL COUNTERS (2)", 20,
            SU_PATH, "root", "wlutil", "counters", NULL);

    run_command("ND OFFLOAD STATUS (2)", 5,
            SU_PATH, "root", "wlutil", "nd_status", NULL);
#endif
    dump_file("INTERRUPTS (2)", "/proc/interrupts");

    print_properties();

    run_command("VOLD DUMP", 10, "vdc", "dump", NULL);
    run_command("SECURE CONTAINERS", 10, "vdc", "asec", "list", NULL

    run_command("FILESYSTEMS & FREE SPACE", 10, "df", NULL);

    run_command("LAST RADIO LOG", 10, "parse_radio_log", "/proc/last

    printf("------ BACKLIGHTS ------\n");
    printf("LCD brightness=");
    dump_file(NULL, "/sys/class/leds/lcd-backlight/brightness");
    printf("Button brightness=");
    dump_file(NULL, "/sys/class/leds/button-backlight/brightness");
    printf("Keyboard brightness=");
    dump_file(NULL, "/sys/class/leds/keyboard-backlight/brightness"
    printf("ALS mode=");
    dump_file(NULL, "/sys/class/leds/lcd-backlight/als");
    printf("LCD driver registers:\n");
    dump_file(NULL, "/sys/class/leds/lcd-backlight/registers");
    printf("\n");

    /* Binder state is expensive to look at as it uses a lot of memo
    dump_file("BINDER FAILED TRANSACTION LOG", "/sys/kernel/debug/bi
    dump_file("BINDER TRANSACTION LOG", "/sys/kernel/debug/binder/tr
    dump_file("BINDER TRANSACTIONS", "/sys/kernel/debug/binder/trans
    dump_file("BINDER STATS", "/sys/kernel/debug/binder/stats");
    dump_file("BINDER STATE", "/sys/kernel/debug/binder/state");

    printf("=========================================================
    printf("== Board\n");
    printf("=========================================================

    dumpstate_board();
    printf("\n");

    /* Migrate the ril_dumpstate to a dumpstate_board()? */
```


海参价格


```
    char ril_dumpstate_timeout[PROPERTY_VALUE_MAX] = {0};
    property_get("ril.dumpstate.timeout", ril_dumpstate_timeout, "30
    if (strnlen(ril_dumpstate_timeout, PROPERTY_VALUE_MAX - 1) > 0)
        if (0 == strncmp(build_type, "user", PROPERTY_VALUE_MAX - 1)
            // su does not exist on user builds, so try running with
            // This way any implementations of vril-dump that do not
            // root can run on user builds.
            run_command("DUMP VENDOR RIL LOGS", atoi(ril_dumpstate_t
                    "vril-dump", NULL);
        } else {
            run_command("DUMP VENDOR RIL LOGS", atoi(ril_dumpstate_t
                    SU_PATH, "root", "vril-dump", NULL);
        }
    }

    printf("=========================================================
    printf("== Android Framework Services\n");
    printf("=========================================================

    /* the full dumpsys is starting to take a long time, so we need
       to increase its timeout.  we really need to do the timeouts i
       dumpsys itself... */
    run_command("DUMPSYS", 60, "dumpsys", NULL);

    printf("=========================================================
    printf("== Checkins\n");
    printf("=========================================================

    run_command("CHECKIN BATTERYSTATS", 30, "dumpsys", "batterystats
    run_command("CHECKIN MEMINFO", 30, "dumpsys", "meminfo", "--chec
    run_command("CHECKIN NETSTATS", 30, "dumpsys", "netstats", "--ch
    run_command("CHECKIN PROCSTATS", 30, "dumpsys", "procstats", "-c
    run_command("CHECKIN USAGESTATS", 30, "dumpsys", "usagestats", "
    run_command("CHECKIN PACKAGE", 30, "dumpsys", "package", "--chec

    printf("=========================================================
    printf("== Running Application Activities\n");
    printf("=========================================================

    run_command("APP ACTIVITIES", 30, "dumpsys", "activity", "all",

    printf("=========================================================
    printf("== Running Application Services\n");
    printf("=========================================================

    run_command("APP SERVICES", 30, "dumpsys", "activity", "service"

```


海参价格

```
    printf("==========================================================
    printf("== Running Application Providers\n");
    printf("==========================================================

    run_command("APP SERVICES", 30, "dumpsys", "activity", "provider


    printf("==========================================================
    printf("== dumpstate: done\n");
    printf("==========================================================
}</code></code></code></code>
```

上面的代码比较长，是因为所要dump的模块太多，但是基本逻辑还是比较清楚的，我们看到基本的数据来源就是：

1.系统属性

2./proc和/sys节点文件

3.执行shell命令获得相关输出

4.logcat输出

5.Android Framework Services信息基本使用dumpsys命令通过binder调用服务中的dump函数获得信息

这里我们需要看一下dumpsys命令的实现，这个命令也是比较简单，实现全部在main函数中：

```
<code class="hljs avrasm"><code class="hljs applescript"><code clas
{
    signal(SIGPIPE, SIG_IGN);
    sp<iservicemanager> sm = defaultServiceManager();
    fflush(stdout);
    if (sm == NULL) {
        ALOGE("Unable to get default service manager!");
        aerr << "dumpsys: Unable to get default service manager!" <<
        return 20;
    }

    Vector<string16> services;
    Vector<string16> args;
    bool showListOnly = false;
    if ((argc == 2) && (strcmp(argv[1], "-l") == 0)) {
        showListOnly = true;
    }
```


海参价格

```
    if ((argc == 1) || showListOnly) {
        services = sm->listServices();
        services.sort(sort_func);
        args.add(String16("-a"));
    } else {
        services.add(String16(argv[1]));
        for (int i=2; i 1) {
        // first print a list of the current services
        aout << "Currently running services:" << endl;

        for (size_t i=0; i<n; ibinder=""> service = sm->checkService(
            if (service != NULL) {
                aout << "  " << services[i] << endl;
            }
        }
    }

    if (showListOnly) {
        return 0;
    }

    for (size_t i=0; i<n; ibinder=""> service = sm->checkService(serv
        if (service != NULL) {
            if (N > 1) {
                aout << "----------------------------------------
                        "-------------------" << endl;
                aout << "DUMP OF SERVICE " << services[i] << ":" << e
            }
            int err = service->dump(STDOUT_FILENO, args);
            if (err != 0) {
                aerr << "Error dumping service info: (" << strerror(e
                        << ") " << services[i] << endl;
            }
        } else {
            aerr << "Can't find service: " << services[i] << endl;
        }
    }

    return 0;
}</n;></n;></argc;></string16></string16></iservicemanager></code></c
```

我们看到它的代码逻辑就是，通过Binder的SM查找参数中的service，然后通过：

```
<code class="hljs avrasm"><code class="hljs applescript"><code class?
```


海参价格

这句来调用service的dump函数。

dumpstate会调用到所有binder中的service的dump函数，因为dumpstate函数执行了这一句：

```
<code class="hljs avrasm"><code class="hljs applescript"><code class?'
   to increase its timeout.  we really need to do the timeouts in
   dumpsys itself... */
run_command("DUMPSYS", 60, "dumpsys", NULL);</code></code></code></coc
```

直接执行dumpsys，没有参数，并且注释中也说的很清楚，就是采集所有的信息。这会执行以下service的dump函数（执行dumpsys | grep “DUMP OF SERVICE”可以看到）：

```
<code class="hljs avrasm">><code class="hljs applescript">><code cla?s
DUMP OF SERVICE SurfaceFlinger:
DUMP OF SERVICE accessibility:
DUMP OF SERVICE account:
DUMP OF SERVICE activity:
DUMP OF SERVICE alarm:
DUMP OF SERVICE android.security.keystore:
DUMP OF SERVICE android.service.gatekeeper.IGateKeeperService:
DUMP OF SERVICE appops:
DUMP OF SERVICE appwidget:
DUMP OF SERVICE assetatlas:
DUMP OF SERVICE audio:
DUMP OF SERVICE backup:
DUMP OF SERVICE battery:
DUMP OF SERVICE batteryproperties:
DUMP OF SERVICE batterystats:
DUMP OF SERVICE bluetooth_manager:
DUMP OF SERVICE carrier_config:
DUMP OF SERVICE clipboard:
DUMP OF SERVICE commontime_management:
DUMP OF SERVICE connectivity:
DUMP OF SERVICE consumer_ir:
DUMP OF SERVICE content:
DUMP OF SERVICE country_detector:
DUMP OF SERVICE cpuinfo:
DUMP OF SERVICE dbinfo:
DUMP OF SERVICE device_policy:
DUMP OF SERVICE deviceidle:
DUMP OF SERVICE devicestoragemonitor:
DUMP OF SERVICE diskstats:
DUMP OF SERVICE display:
DUMP OF SERVICE display.qservice:
```


海参价格

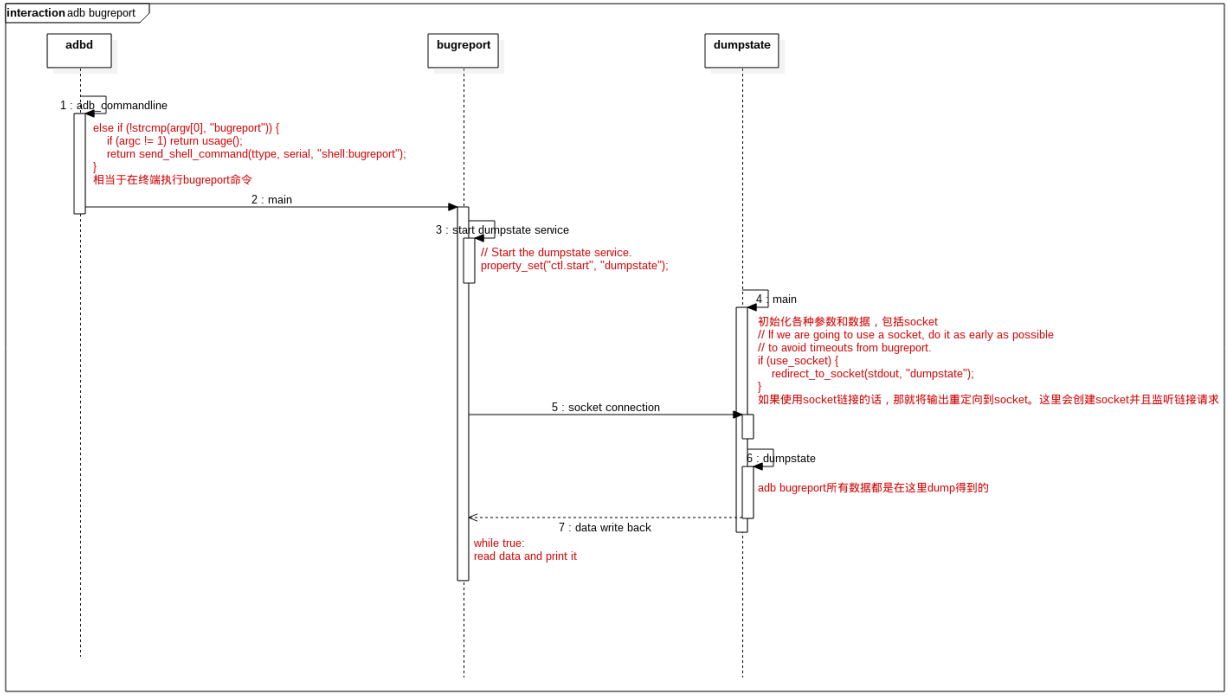

```
DUMP OF SERVICE dreams:
DUMP OF SERVICE drm.drmManager:
DUMP OF SERVICE dropbox:
DUMP OF SERVICE ethernet:
DUMP OF SERVICE fingerprint:
DUMP OF SERVICE gfxinfo:
DUMP OF SERVICE graphicsstats:
DUMP OF SERVICE imms:
DUMP OF SERVICE input:
DUMP OF SERVICE input_method:
DUMP OF SERVICE iphonesubinfo:
DUMP OF SERVICE isms:
DUMP OF SERVICE isub:
DUMP OF SERVICE jobscheduler:
DUMP OF SERVICE launcherapps:
DUMP OF SERVICE location:
DUMP OF SERVICE lock_settings:
DUMP OF SERVICE media.audio_flinger:
DUMP OF SERVICE media.audio_policy:
DUMP OF SERVICE media.camera:
DUMP OF SERVICE media.camera.proxy:
DUMP OF SERVICE media.player:
DUMP OF SERVICE media.radio:
DUMP OF SERVICE media.resource_manager:
DUMP OF SERVICE media.sound_trigger_hw:
DUMP OF SERVICE media_projection:
DUMP OF SERVICE media_router:
DUMP OF SERVICE media_session:
DUMP OF SERVICE meminfo:
DUMP OF SERVICE midi:
DUMP OF SERVICE mount:
DUMP OF SERVICE netpolicy:
DUMP OF SERVICE netstats:
DUMP OF SERVICE network_management:
DUMP OF SERVICE network_score:
DUMP OF SERVICE nfc:
DUMP OF SERVICE notification:
DUMP OF SERVICE package:
DUMP OF SERVICE permission:
DUMP OF SERVICE persistent_data_block:
DUMP OF SERVICE phone:
DUMP OF SERVICE power:
DUMP OF SERVICE print:
DUMP OF SERVICE processinfo:
DUMP OF SERVICE procstats:
DUMP OF SERVICE restrictions:
DUMP OF SERVICE rttmanager:
```


海参价格

```
DUMP OF SERVICE samplingprofiler:
DUMP OF SERVICE scheduling_policy:
DUMP OF SERVICE search:
DUMP OF SERVICE sensorservice:
DUMP OF SERVICE serial:
DUMP OF SERVICE servicediscovery:
DUMP OF SERVICE simphonebook:
DUMP OF SERVICE sip:
DUMP OF SERVICE statusbar:
DUMP OF SERVICE telecom:
DUMP OF SERVICE telephony.registry:
DUMP OF SERVICE textservices:
DUMP OF SERVICE trust:
DUMP OF SERVICE uimode:
DUMP OF SERVICE updatelock:
DUMP OF SERVICE usagestats:
DUMP OF SERVICE usb:
DUMP OF SERVICE user:
DUMP OF SERVICE vibrator:
DUMP OF SERVICE voiceinteraction:
DUMP OF SERVICE wallpaper:
DUMP OF SERVICE webviewupdate:
DUMP OF SERVICE wifi:
DUMP OF SERVICE wifip2p:
DUMP OF SERVICE wifiscanner:
DUMP OF SERVICE window:</code></code></code></code></code></code></c
```

这里总结以下，上面的bugreport整体逻辑如下图描述（如果图片太小看不清，请下载图片并查看）：


海参价格



adb bugreport的其他选项

bugreport本身并没有什么选项，主要是通过dumpsys等命令配合完成，详见battery historian项目主页：https://github.com/google/battery-historian，以下是个总结：

1）. 重置电池统计信息：

```
<code class="hljs avrasm"><code class="hljs applescript"><code class?'
```

2）. Wakelock analysis全部wakelock信息：

```
<code class="hljs avrasm"><code class="hljs applescript"><code class?'
```

3）. Kernel trace analysis分析内核，主要分析wakeup source和wakelock activities，首先使能kernel分析：


海参价格

```
<code class="hljs avrasm"><code class="hljs applescript"><code clas?
$ adb shell

# Set the events to trace.
$ echo "power:wakeup_source_activate" >> /d/tracing/set_event
$ echo "power:wakeup_source_deactivate" >> /d/tracing/set_event

# The default trace size for most devices is 1MB, which is relatively
# 8MB to 10MB should be a decent size for 5-6 hours of logging.

$ echo 8192 > /d/tracing/buffer_size_kb

$ echo 1 > /d/tracing/tracing_on</code></code></code></code></code></
```

然后获得log：

```
<code class="hljs avrasm"><code class="hljs applescript"><code class?'
$ adb pull /d/tracing/trace <some path="">

# Take a bug report at this time.
$ adb bugreport > bugreport.txt</some></code></code></code></code></co
```



海参价格



灵芝价格


虫草价格表


透明手机售价

移动靓号


海参价格

点击复制链接 与好友分享!　回本站首页

相关TAG标签　工具

上一篇：android视频播放的代码

下一篇：Android ORM数据库之GreenDao使用教程及源码分析

相关文章

使用git工具下载android.jar Source
Android学习笔记Android必备开发工具之
developer盘点Android开发者必备的十大
安装Android开发工具
Android实用工具Hierarchy Viewer实战
Android手机归属地查询工具
【Android游戏开发之十】（优化处理）
【Android游戏开发十一】手把手让你爱
android里图片下载工具类AsyncImageLo
Android深入浅出系列之Android工具的使

热门专题推荐 python div+css css教程 html5 html教程 jquery Android SDK
php mysql oracle

海参价格

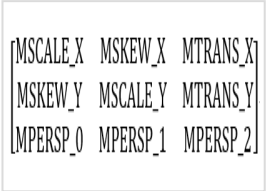
图文推荐


润滑剂怎么使用

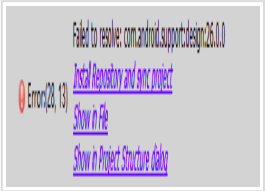

灵芝价格


海参价格表

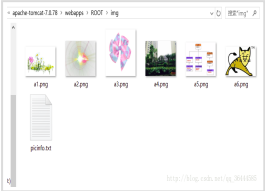

牙齿黄怎么变白

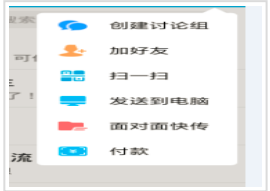


Android Matrix的用



Failed to resolve


安卓开发-手机上显示



Android开发技术学习

登录

来说两句吧...

还没有评论，快来抢沙发吧！

红黑联盟正在使用畅言

海参价格

国产机械表排名 家用小电梯价格 短信验证码平台 5g手机 5G手机 小程序开发公司 明天股市走势预测
世界名表排行榜 affinity 移动不限流量卡 金士顿存储卡 开发一个app多少钱 便宜手机大全

证书+能力 安全工程师 软件工程师 网站工程师 网络工程师 电脑工程师
为新手量身定做的课程，让菜鸟快速变身高手 正规公司助您腾飞
不断增加新科目
立即加入

版权所有: 红黑联盟--致力于做实用的IT技术学习网站