**Android中dumpsys命令用法简单介绍**

 更新时间：2017年03月06日 11:38:48   投稿：lqh   [[https://www.jb51.net/skin/2018/images/text-message.png](https://www.jb51.net/article/107568.htm#comments) 我要评论](https://www.jb51.net/article/107568.htm#comments)

这篇文章主要介绍了Android中dumpsys命令用法简单介绍的相关资料,需要的朋友可以参考下

在Android手机上， 通过使用adb shell命令可以进入android系统的shell， 该shell除支持一些常用的标准命令之外，还支持一些和android系统相关的其他命令， 这些命令可以打印出系统当前的状态信息。 dumpsys就是这样一个命令。

使用 adb shell 进入命令行， 敲入dumpsys， 打印出的信息超级多, 在我的nexus 5机器上，输出多达67000行。 首先从打印信息的开始部分看起：

Currently running services:

SurfaceFlinger

accessibility

account

activity

alarm

android.security.keystore

appops

appwidget

assetatlas

audio

backup

battery

batterypropreg

batterystats

bluetooth\_manager

clipboard

commontime\_management

connectivity

consumer\_ir

content

country\_detector

cpuinfo

dbinfo

device\_policy

devicestoragemonitor

diskstats

display

display.qservice

dreams

drm.drmManager

dropbox

entropy

gfxinfo

hardware

input

input\_method

iphonesubinfo

isms

location

lock\_settings

media.audio\_flinger

media.audio\_policy

media.camera

media.player

media\_router

meminfo

mount

netpolicy

netstats

network\_management

nfc

notification

package

permission

phone

power

print

procstats

samplingprofiler

scheduling\_policy

search

sensorservice

serial

servicediscovery

simphonebook

sip

statusbar

telephony.registry

textservices

uimode

updatelock

usagestats

usb

user

vibrator

wallpaper

wifi

wifip2p

window

由此可知，该命令打印出的是系统当前的所有正在运行的服务的信息。从中可以看到我们常见的一些服务。 activity表示的是ActivityManagerService,用于管理应用程序的四大组件， appops表示的是AppOpsManagerService, 用于管理和配置权限。window表示的是WindowManagerService, 用于管理系统中的窗口， account表示的是AccountManagerService，用于管理系统中的账户信息。

在输出信息中， 紧接着的是每个服务当前的详细状态信息，每个服务的状态信息一般都比较多， 下面列举一个信息较少的服务信息：

-------------------------------------------------------------------------------

DUMP OF SERVICE accessibility:

ACCESSIBILITY MANAGER (dumpsys accessibility)

User state[attributes:{id=0, currentUser=true, accessibilityEnabled=false, touchExplorationEnabled=false, displayMagnificationEnabled=false}

services:{}]

-------------------------------------------------------------------------------

每个服务信息之间使用横线间隔。

在开发过程中， 使用dumpsys的最常用的用途是查看ActivityManagerService服务的信息。 下面重点分析ActivityManagerService。 该服务的信息从下面一行开始:

-------------------------------------------------------------------------------

DUMP OF SERVICE activity:

我们知道android应用程序的四大组件中有三个是由intent来驱动的。所以在ActivityManagerService的信息中，专门有一部分来记录当前待处理的Intent， 称为pending intent 。格式如下：

ACTIVITY MANAGER PENDING INTENTS (dumpsys activity intents)

\* PendingIntentRecord{65ced5f0 com.baidu.BaiduMap broadcastIntent}

uid=10084 packageName=com.baidu.BaiduMap type=broadcastIntent flags=0x0

requestIntent=act=com.baidu.locTest.LocationServer4.1.8

sent=true canceled=false

\* PendingIntentRecord{66412b40 com.android.phone broadcastIntent}

uid=1001 packageName=com.android.phone type=broadcastIntent flags=0x0

requestIntent=act=com.android.phone.ACTION\_CALL\_BACK\_FROM\_NOTIFICATION dat=tel:153%201571%209213 cmp=com.android.phone/.PhoneGlobals$NotificationBroadcastReceiver

直接在shell中键入dumpsys activity intents， 会直接得到这些信息。 每个待处理的intent在ActivityManagerService内部使用一个PendingIntentRecord对象表示， 并且会显示出所在应用的包名， 并且会指定该intent是用来发送广播， 启动activity或是启动service。

下面的信息是系统中的broadcast receiver的状态信息， 从下面一行开始：

-------------------------------------------------------------------------------

ACTIVITY MANAGER BROADCAST STATE (dumpsys activity broadcasts)

直接在shell中键入umpsys activity broadcasts, 会直接得到这部分信息。

首先会列出系统中所有注册的广播接收者， 格式如下：

Registered Receivers:

\* ReceiverList{660f2e40 857 com.android.systemui/10012/u-1 remote:660f2ca0}

app=857:com.android.systemui/u0a12 pid=857 uid=10012 user=-1

Filter #0: BroadcastFilter{660f2ea0}

Action: "android.intent.action.SHOW\_BRIGHTNESS\_DIALOG"

\* ReceiverList{66045cf0 3867 com.meitu.meiyancamera:pushservice/10091/u0 remote:65e756c8}

app=3867:com.meitu.meiyancamera:pushservice/u0a91 pid=3867 uid=10091 user=0

Filter #0: BroadcastFilter{66045d50}

Action: "android.net.conn.CONNECTIVITY\_CHANGE"

之后列出的是位于前台的历史广播， 格式如下：

Historical broadcasts [foreground]:

Historical Broadcast foreground #0:

BroadcastRecord{65610478 u-1 android.intent.action.TIME\_TICK} to user -1

Intent { act=android.intent.action.TIME\_TICK flg=0x50000014 (has extras) }

extras: Bundle[{android.intent.extra.ALARM\_COUNT=1}]

caller=android null pid=-1 uid=1000

dispatchClockTime=Tue Oct 07 10:53:00 GMT+08:00 2014

dispatchTime=-14s534ms finishTime=-14s436ms

resultTo=null resultCode=0 resultData=null

resultAbort=false ordered=true sticky=false initialSticky=false

nextReceiver=10 receiver=null

Receiver #0: BroadcastFilter{6568af00 u0 ReceiverList{65692db8 748 system/1000/u0 local:65b2a730}}

Receiver #1: BroadcastFilter{660c8768 u0 ReceiverList{660c81a8 857 com.android.systemui/10012/u0 remote:660c78c0}}

Receiver #2: BroadcastFilter{65d2d230 u0 ReceiverList{65d2cb20 857 com.android.systemui/10012/u0 remote:65a8bb70}}

Receiver #3: BroadcastFilter{65f042e8 u0 ReceiverList{65f04d90 857 com.android.systemui/10012/u0 remote:65f9f2c0}}

Receiver #4: BroadcastFilter{65a9d5f0 u0 ReceiverList{65affbc8 857 com.android.systemui/10012/u0 remote:65b41d50}}

Receiver #5: BroadcastFilter{6682be08 u0 ReceiverList{6682bda8 857 com.android.systemui/10012/u0 remote:665c2478}}

Receiver #6: BroadcastFilter{667a6c48 u0 ReceiverList{6671f128 857 com.android.systemui/10012/u0 remote:662c4048}}

Receiver #7: BroadcastFilter{66901318 u0 ReceiverList{66949870 11260 com.youku.phone:MMS/10080/u0 remote:6695e528}}

Receiver #8: BroadcastFilter{663a4b60 u0 ReceiverList{663a5148 857 com.android.systemui/10012/u0 remote:663c3530}}

Receiver #9: BroadcastFilter{6633df98 u0 ReceiverList{6633f628 857 com.android.systemui/10012/u0 remote:66370860}}

然后是位于后台的历史广播，格式如下：

Historical broadcasts [background]:

Historical Broadcast background #0:

BroadcastRecord{66430db0 u0 cn.etouch.ecalendar\_CN.ETOUCH.ECALENDAR.WEATHER\_HAS\_UPDATE} to user 0

Intent { act=cn.etouch.ecalendar\_CN.ETOUCH.ECALENDAR.WEATHER\_HAS\_UPDATE flg=0x10 (has extras) }

extras: Bundle[mParcelledData.dataSize=44]

caller=cn.etouch.ecalendar 3036:cn.etouch.ecalendar:remote/u0a140 pid=3036 uid=10140

dispatchClockTime=Tue Oct 07 10:53:00 GMT+08:00 2014

dispatchTime=-14s475ms finishTime=-14s400ms

resultTo=null resultCode=0 resultData=null

nextReceiver=5 receiver=null

Receiver #0: ResolveInfo{66335e40 cn.etouch.ecalendar/.widget.myWidget\_weather p=1000 m=0x108000}

priority=1000 preferredOrder=0 match=0x108000 specificIndex=-1 isDefault=false

ActivityInfo:

name=cn.etouch.ecalendar.widget.myWidget\_weather

packageName=cn.etouch.ecalendar

labelRes=0x7f06016c nonLocalizedLabel=null icon=0x0

enabled=true exported=true processName=cn.etouch.ecalendar

taskAffinity=cn.etouch.ecalendar targetActivity=null

ApplicationInfo:

packageName=cn.etouch.ecalendar

labelRes=0x7f060042 nonLocalizedLabel=null icon=0x7f0200c5

className=cn.etouch.ecalendar.common.ApplicationManager

processName=cn.etouch.ecalendar

taskAffinity=cn.etouch.ecalendar

uid=10140 flags=0x88be44 theme=0x7f0a0029

requiresSmallestWidthDp=0 compatibleWidthLimitDp=0 largestWidthLimitDp=0

sourceDir=/data/app/cn.etouch.ecalendar-1.apk

seinfo=default

dataDir=/data/data/cn.etouch.ecalendar

enabled=true targetSdkVersion=11

supportsRtl=false

下面是provider相关的信息， 格式如下：

-------------------------------------------------------------------------------

ACTIVITY MANAGER CONTENT PROVIDERS (dumpsys activity providers)

Published single-user content providers (by class):

\* ContentProviderRecord{65cd5f60 u0 com.redbend.vdmc/com.redbend.others.NodeProvider}

package=com.redbend.vdmc process=com.redbend.vdmc

proc=ProcessRecord{65d540d8 1018:com.redbend.vdmc/1001}

uid=1001 provider=android.content.ContentProviderProxy@658b6690

singleton=true

authority=com.lge.node

\* ContentProviderRecord{65d1b150 u0 com.android.phone/.IccProvider}

package=com.android.phone process=com.android.phone

proc=ProcessRecord{65fe2488 1000:com.android.phone/1001}

uid=1001 provider=android.content.ContentProviderProxy@660cbf38

singleton=true

authority=icc

isSyncable=false multiprocess=true initOrder=0

直接在命令行键入dumpsys activity providers，可以直接得到provider相关的信息。由输出信息可知， 在AtivityManagerService中，一个provider组件使用一个ContentProviderRecord对象表示。

服务（Service）相关的信息如下：

-------------------------------------------------------------------------------

ACTIVITY MANAGER SERVICES (dumpsys activity services)

User 0 active services:

\* ServiceRecord{660da0f0 u0 com.android.bluetooth/.hid.HidService}

intent={act=android.bluetooth.IBluetoothInputDevice cmp=com.android.bluetooth/.hid.HidService}

packageName=com.android.bluetooth

processName=com.android.bluetooth

baseDir=/system/app/Bluetooth.apk

dataDir=/data/data/com.android.bluetooth

app=null

createTime=-1d16h27m16s523ms startingBgTimeout=--

lastActivity=-17h20m54s385ms restartTime=-- createdFromFg=false

Bindings:

\* IntentBindRecord{660da3e0}:

intent={act=android.bluetooth.IBluetoothInputDevice cmp=com.android.bluetooth/.hid.HidService}

binder=null

requested=false received=false hasBound=false doRebind=false

在shell中直接键入dumpsys activity services命令， 可以直接得到Service信息。

Activity相关的信息格式如下：

-------------------------------------------------------------------------------

ACTIVITY MANAGER ACTIVITIES (dumpsys activity activities)

Stack #0:

Task id #1

\* TaskRecord{65cc6860 #1 A=com.google.android.googlequicksearchbox U=0 sz=1}

numActivities=1 rootWasReset=true userId=0 mTaskType=1 numFullscreen=1 mOnTopOfHome=true

affinity=com.google.android.googlequicksearchbox

intent={act=android.intent.action.MAIN cat=[android.intent.category.HOME] flg=0x10600000 cmp=com.google.android.googlequicksearchbox/com.google.android.launcher.GEL}

realActivity=com.google.android.googlequicksearchbox/com.google.android.launcher.GEL

Activities=[ActivityRecord{66020ee8 u0 com.google.android.googlequicksearchbox/com.google.android.launcher.GEL t1}]

askedCompatMode=false

lastThumbnail=android.graphics.Bitmap@6653bf90 lastDescription=null

lastActiveTime=144235207 (inactive for 1424s)

\* Hist #0: ActivityRecord{66020ee8 u0 com.google.android.googlequicksearchbox/com.google.android.launcher.GEL t1}

packageName=com.google.android.googlequicksearchbox processName=com.google.android.googlequicksearchbox

launchedFromUid=0 launchedFromPackage=null userId=0

app=ProcessRecord{66105ac8 1058:com.google.android.googlequicksearchbox/u0a19}

Intent { act=android.intent.action.MAIN cat=[android.intent.category.HOME] flg=0x10000000 cmp=com.google.android.googlequicksearchbox/com.google.android.launcher.GEL }

frontOfTask=true task=TaskRecord{65cc6860 #1 A=com.google.android.googlequicksearchbox U=0 sz=1}

taskAffinity=com.google.android.googlequicksearchbox

realActivity=com.google.android.googlequicksearchbox/com.google.android.launcher.GEL

baseDir=/system/priv-app/Velvet.apk

dataDir=/data/user/0/com.google.android.googlequicksearchbox

stateNotNeeded=true componentSpecified=false mActivityType=1

compat={480dpi} labelRes=0x7f0a04c5 icon=0x7f030002 theme=0x7f0e000d

config={1.0 460mcc1mnc zh\_CN ldltr sw360dp w360dp h567dp 480dpi nrml port finger -keyb/v/h -nav/h s.11}

launchFailed=false launchCount=0 lastLaunchTime=-17h20m56s529ms

haveState=true icicle=Bundle[mParcelledData.dataSize=12720]

state=STOPPED stopped=true delayedResume=false finishing=false

keysPaused=false inHistory=true visible=false sleeping=true idle=true

fullscreen=true noDisplay=false immersive=false launchMode=2

frozenBeforeDestroy=false thumbnailNeeded=false forceNewConfig=false

mActivityType=HOME\_ACTIVITY\_TYPE

thumbHolder: 65cc6860 bm=android.graphics.Bitmap@6653bf90 desc=null

waitingVisible=false nowVisible=false lastVisibleTime=-21m53s598ms

Task id #24

\* TaskRecord{667de460 #24 A=com.android.systemui U=0 sz=1}

numActivities=1 rootWasReset=false userId=0 mTaskType=2 numFullscreen=1 mOnTopOfHome=true

affinity=com.android.systemui

intent={act=com.android.systemui.recent.action.TOGGLE\_RECENTS flg=0x10c00000 cmp=com.android.systemui/.recent.RecentsActivity}

realActivity=com.android.systemui/.recent.RecentsActivity

Activities=[ActivityRecord{6561e970 u0 com.android.systemui/.recent.RecentsActivity t24}]

askedCompatMode=false

lastThumbnail=android.graphics.Bitmap@662b3118 lastDescription=null

lastActiveTime=87755110 (inactive for 57904s)

\* Hist #0: ActivityRecord{6561e970 u0 com.android.systemui/.recent.RecentsActivity t24}

packageName=com.android.systemui processName=com.android.systemui

launchedFromUid=10012 launchedFromPackage=com.android.systemui userId=0

app=ProcessRecord{6602d698 857:com.android.systemui/u0a12}

Intent { act=com.android.systemui.recent.action.TOGGLE\_RECENTS flg=0x10800000 cmp=com.android.systemui/.recent.RecentsActivity }

frontOfTask=true task=TaskRecord{667de460 #24 A=com.android.systemui U=0 sz=1}

taskAffinity=com.android.systemui

realActivity=com.android.systemui/.recent.RecentsActivity

baseDir=/system/priv-app/SystemUI.apk

dataDir=/data/data/com.android.systemui

stateNotNeeded=false componentSpecified=true mActivityType=2

compat={480dpi} labelRes=0x7f0a0080 icon=0x10804a5 theme=0x7f0e0000

config={1.0 460mcc1mnc zh\_CN ldltr sw360dp w360dp h567dp 480dpi nrml port finger -keyb/v/h -nav/h s.9}

launchFailed=false launchCount=0 lastLaunchTime=-9h59m21s658ms

haveState=true icicle=Bundle[mParcelledData.dataSize=44]

state=STOPPED stopped=true delayedResume=false finishing=false

keysPaused=false inHistory=true visible=false sleeping=true idle=true

fullscreen=true noDisplay=false immersive=false launchMode=3

frozenBeforeDestroy=false thumbnailNeeded=false forceNewConfig=false

mActivityType=RECENTS\_ACTIVITY\_TYPE

thumbHolder: 667de460 bm=android.graphics.Bitmap@662b3118 desc=null

waitingVisible=false nowVisible=false lastVisibleTime=-7h8m31s600ms

Running activities (most recent first):

TaskRecord{65cc6860 #1 A=com.google.android.googlequicksearchbox U=0 sz=1}

Run #1: ActivityRecord{66020ee8 u0 com.google.android.googlequicksearchbox/com.google.android.launcher.GEL t1}

TaskRecord{667de460 #24 A=com.android.systemui U=0 sz=1}

Run #0: ActivityRecord{6561e970 u0 com.android.systemui/.recent.RecentsActivity t24}

mLastPausedActivity: ActivityRecord{66020ee8 u0 com.google.android.googlequicksearchbox/com.google.android.launcher.GEL t1}

在shell中直接键入dumpsys activity activities, 可以直接得到activity的相关信息。

和其他组件相比， Activity的管理比较特殊。 Activity是以栈的形式管理的， 每个栈中存在若干个任务（task）， 每个任务又由若干个Activity组成。

上面简单介绍了ActivityManagerService中的四大组件相关的信息。 关于其他服务中的信息， 由于信息量太大， 无法一一介绍， 请读者自行查看。

感谢阅读，希望能帮助到大家，谢谢大家对本站的支持！

**Android偏好设置及findPreferredActivity分析**

2016年08月20日 17:31:12 [aaajj](https://me.csdn.net/aaajj" \t "_blank) 阅读数：2263

在手机使用中，经常需要设置默认打开方式，比如安装了好几个浏览器的时候，在打开网页链接的时候，在第一次会弹框让用户进行应用的选择。

这里以浏览器为例进行说明。

如果之前设置了默认浏览器，再安装另外一个浏览器后，再次打开链接，会重新弹框要求用户选择打开方式，这样设计的目的是让新安装的应用有机会得到用户的选择使用。

在之前的文章中，已经知道了存储偏好设置的文件是package-restrictions.xml

现在来查看下package-restrictions.xml，把UC设置为默认浏览器

package-restrictions.xml

        <item name="com.UCMobile/.main.UCMobile" match="200000" always="true" set="2">

            <set name="com.UCMobile/.main.UCMobile" />

            <set name="com.android.browser/.BrowserActivity" />

            <filter>

                <action name="android.intent.action.VIEW" />

                <cat name="android.intent.category.DEFAULT" />

                <scheme name="http" />

            </filter>

        </item>

安装另外一个浏览器后，调用到findPreferredActivity

                            if (always && !pa.mPref.sameSet(query)) {

                                Slog.d(TAG, "===== pa.mPref.Set =" + pa.mPref.mSetPackages);

                                for (String astr : pa.mPref.mSetPackages)

                                {

                                        Slog.d(TAG, "set[]=" + astr);

                                }

                                Slog.i(TAG, "Result set changed, dropping preferred activity for "

                                        + intent + " type " + resolvedType);

                                if (DEBUG\_PREFERRED) {

                                    Slog.v(TAG, "Removing preferred activity since set changed "

                                            + pa.mPref.mComponent);

                                }

pa.mPref.mSetPackages 为package-restrictions.xml里set的内容，如

            <set name="com.UCMobile/.main.UCMobile" />

            <set name="com.android.browser/.BrowserActivity" />

query为实时从pms里查询得到的满足条件的activity相关信息

当发现两个数据不一致的时候，就认为系统里的ap发生了变化，走到

Slog.i(TAG, "Result set changed, dropping preferred activity for "

                                        + intent + " type " + resolvedType);

删除掉之前的默认设置。然后由

ResolverActivity去处理，显示选择界面

基本处理流程是

点击url网络连接 -> findPreferredActivity, 查看是否有默认打开方式

如果有，直接打开

如果没有，

ResolverActivity去处理，显示选择界面，这个时候，选择“总是”（always），

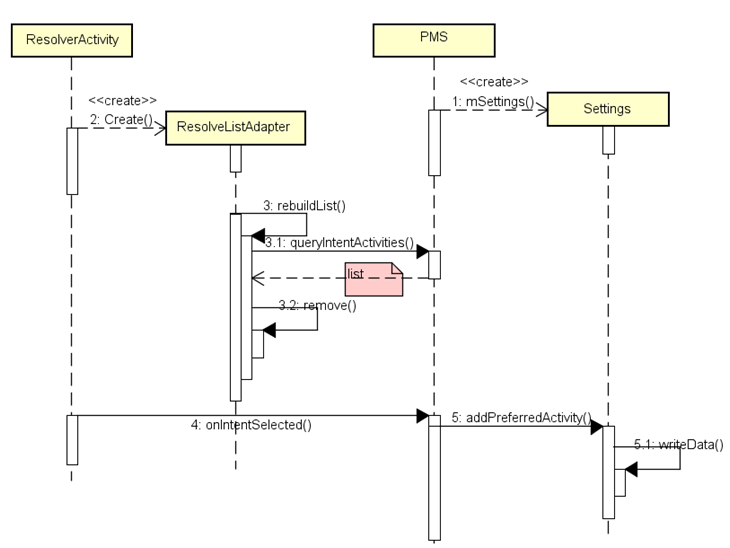
ResolverActivity

调用

 pm.addPreferredActivity(filter, bestMatch, set, intent.getComponent());

去重新设置偏好。

可参考下图



值得注意的是，

ResolverActivity里对activity的

priority进行了处理，如果一个ap注册的时候，把

intent-filter里的

priority写为了负数，如下所示

<intent-filter android:priority="-1" >

 <category android:name="android.intent.category.DEFAULT" />

 <action android:name="android.intent.action.VIEW" />

 <data android:scheme="http" />

那么，在选择默认浏览器的时候，就不会显示出这个应用。

因为在

private void rebuildList() 里

有这样的处理

        for (int i=1; i<N; i++) {

                    ResolveInfo ri = currentResolveList.get(i);

                    if (DEBUG) Log.v(

                        TAG,

                        r0.activityInfo.name + "=" +

                        r0.priority + "/" + r0.isDefault + " vs " +

                        ri.activityInfo.name + "=" +

                        ri.priority + "/" + ri.isDefault);

                    if (r0.priority != ri.priority ||

                        r0.isDefault != ri.isDefault) {

                        while (i < N) {

                            if (mOrigResolveList == currentResolveList) {

                                mOrigResolveList = new ArrayList<ResolveInfo>(mOrigResolveList);

                            }

                            currentResolveList.remove(i);

导致

priority为负数的不会显示出来。

了解了这些，就可以去分析这个问题

<http://www.zhihu.com/question/27313953/answer/36308142>

在6.0的版本上，对

priority

的处理有些变化，比如对于自定义的ap  Test，

priority

设置为大于0的数

<intent-filter android:priority="3" >

 <category android:name="android.intent.category.DEFAULT" />

 <action android:name="android.intent.action.VIEW" />

 <data android:scheme="http" />

之前手机上安装了3个浏览器（

priority都为0的

），其中UC设置为了默认浏览器，再安装Test，

接着打开网络链接，会弹出选择界面，

在L版本上会列出所有的浏览器，M上呢？

答案是只显示出UC和Test

因为M里的做了些修改，查看PackageManagerService.java的

queryIntentActivities可以很容易的找到

附

package-restrictions.xml

        <item name="com.UCMobile/.main.UCMobile" match="200000" always="true" set="2">

            <set name="com.UCMobile/.main.UCMobile" />

            <set name="com.android.browser/.BrowserActivity" />

            <filter>

                <action name="android.intent.action.VIEW" />

                <cat name="android.intent.category.DEFAULT" />

                <scheme name="http" />

            </filter>

        </item>

里match的来源

这个查看过程非常复杂，

这里直接说下结果，是IntentFilter.java里的matchData方法

附相关log，注意，pms里的DEBUG\_PREFERRED开关需要打开

05-01 08:53:56.290: D/PackageManager(918): ===== findPreferredActivity called

05-01 08:53:56.290: D/PackageManager(918): ===== begin =====

05-01 08:53:56.290: D/PackageManager(918): com.android.server.pm.PackageManagerService.findPreferredActivity(PackageManagerService.java:3575)

05-01 08:53:56.290: D/PackageManager(918): com.android.server.pm.PackageManagerService.chooseBestActivity(PackageManagerService.java:3494)

05-01 08:53:56.290: D/PackageManager(918): com.android.server.pm.PackageManagerService.resolveIntent(PackageManagerService.java:3432)

05-01 08:53:56.290: D/PackageManager(918): android.content.pm.IPackageManager$Stub.onTransact(IPackageManager.java:507)

05-01 08:53:56.290: D/PackageManager(918): com.android.server.pm.PackageManagerService.onTransact(PackageManagerService.java:2264)

05-01 08:53:56.290: D/PackageManager(918): android.os.Binder.execTransact(Binder.java:451)

05-01 08:53:56.290: V/PackageManager(918): Looking for presistent preferred activities...

05-01 08:53:56.290: V/PackageManager(918): Looking for preferred activities...

05-01 08:53:56.290: D/IntentResolver(918): ===== queryIntent called

05-01 08:53:56.290: V/IntentResolver(918): Resolving type=null scheme=http defaultOnly=true userId=0 of Intent { act=android.intent.action.VIEW dat=[http://wap.uc.cn](http://wap.uc.cn/) flg=0x8 }

05-01 08:53:56.290: V/IntentResolver(918): Scheme list: [PreferredActivity{0x43bb27d com.UCMobile/.main.UCMobile}, null]

05-01 08:53:56.290: D/IntentResolver(918): ===== buildResolveList called

05-01 08:53:56.291: D/IntentResolver(918): ===== begin =====

05-01 08:53:56.291: D/IntentResolver(918): com.android.server.IntentResolver.buildResolveList(IntentResolver.java:677)

05-01 08:53:56.291: D/IntentResolver(918): com.android.server.IntentResolver.queryIntent(IntentResolver.java:458)

05-01 08:53:56.291: D/IntentResolver(918): com.android.server.pm.PackageManagerService.findPreferredActivity(PackageManagerService.java:3597)

05-01 08:53:56.291: D/IntentResolver(918): com.android.server.pm.PackageManagerService.chooseBestActivity(PackageManagerService.java:3494)

05-01 08:53:56.291: D/IntentResolver(918): com.android.server.pm.PackageManagerService.resolveIntent(PackageManagerService.java:3432)

05-01 08:53:56.291: D/IntentResolver(918): android.content.pm.IPackageManager$Stub.onTransact(IPackageManager.java:507)

05-01 08:53:56.291: D/IntentResolver(918): com.android.server.pm.PackageManagerService.onTransact(PackageManagerService.java:2264)

05-01 08:53:56.291: D/IntentResolver(918): android.os.Binder.execTransact(Binder.java:451)

05-01 08:53:56.291: V/IntentResolver(918): Matching against filter PreferredActivity{0x43bb27d com.UCMobile/.main.UCMobile}

05-01 08:53:56.291: D/IntentFilter(918): ===== matchData called

05-01 08:53:56.291: D/IntentFilter(918): ===== match=200000

05-01 08:53:56.291: V/IntentResolver(918):   Filter matched!  match=0x208000 hasDefault=true

05-01 08:53:56.291: V/IntentResolver(918):     43bb27d com.UCMobile/.main.UCMobile

05-01 08:53:56.292: V/IntentResolver(918):      mMatch=0x200000 mAlways=true

05-01 08:53:56.292: V/IntentResolver(918):       Selected from:

05-01 08:53:56.292: V/IntentResolver(918):         com.UCMobile/.main.UCMobile

05-01 08:53:56.292: V/IntentResolver(918):         com.android.browser/.BrowserActivity

05-01 08:53:56.292: V/IntentResolver(918):     Action: "android.intent.action.VIEW"

05-01 08:53:56.292: V/IntentResolver(918):     Category: "android.intent.category.DEFAULT"

05-01 08:53:56.292: V/IntentResolver(918):     Scheme: "http"

05-01 08:53:56.292: V/IntentResolver(918): Final result list:

05-01 08:53:56.292: V/IntentResolver(918):   PreferredActivity{0x43bb27d com.UCMobile/.main.UCMobile}

05-01 08:53:56.292: V/PackageManager(918): Figuring out best match...

05-01 08:53:56.292: V/PackageManager(918): Match for ActivityInfo{21f95286 com.android.browser.BrowserActivity}: 0x0

05-01 08:53:56.292: V/PackageManager(918): Match for ActivityInfo{7f12747 com.UCMobile.main.UCMobile}: 0x208000

05-01 08:53:56.292: V/PackageManager(918): Match for ActivityInfo{38f8b74 com.taobao.browser.BrowserActivity}: 0x208000

05-01 08:53:56.292: D/PackageManager(918): ===== query list is:

05-01 08:53:56.292: D/PackageManager(918): ===== ResolveInfo{2e13ff6b com.android.browser/.BrowserActivity m=0x208000}

05-01 08:53:56.292: D/PackageManager(918): ===== ResolveInfo{2e188ac8 com.UCMobile/.main.UCMobile m=0x208000}

05-01 08:53:56.292: D/ResolveInfo(918): tos ===== priority=-1

05-01 08:53:56.292: D/PackageManager(918): ===== ResolveInfo{396c3761 com.taobao.taobao/com.taobao.browser.BrowserActivity p=-1 m=0x208000}

05-01 08:53:56.292: V/PackageManager(918): Best match: 0x208000

05-01 08:53:56.292: V/PackageManager(918): Checking PreferredActivity ds=http

05-01 08:53:56.292: V/PackageManager(918):   component=ComponentInfo{com.UCMobile/com.UCMobile.main.UCMobile}

05-01 08:53:56.292: V/PackageManager(918):   Action: "android.intent.action.VIEW"

05-01 08:53:56.292: V/PackageManager(918):   Category: "android.intent.category.DEFAULT"

05-01 08:53:56.292: V/PackageManager(918):   Scheme: "http"

05-01 08:53:56.292: V/PackageManager(918): Found preferred activity:

05-01 08:53:56.292: V/PackageManager(918):   name=com.UCMobile.main.UCMobile

05-01 08:53:56.292: V/PackageManager(918):   packageName=com.UCMobile

05-01 08:53:56.292: V/PackageManager(918):   labelRes=0x7f070004 nonLocalizedLabel=null icon=0x0 banner=0x0

05-01 08:53:56.292: V/PackageManager(918):   enabled=true exported=true processName=com.UCMobile

05-01 08:53:56.292: V/PackageManager(918):   taskAffinity=com.UCMobile targetActivity=null persistableMode=PERSIST\_ROOT\_ONLY

05-01 08:53:56.292: V/PackageManager(918):   launchMode=2 flags=0x8a theme=0x103000f

05-01 08:53:56.292: V/PackageManager(918):   screenOrientation=-1 configChanges=0x400004a3 softInputMode=0x20

05-01 08:53:56.292: V/PackageManager(918):   ApplicationInfo:

05-01 08:53:56.292: V/PackageManager(918):     packageName=com.UCMobile

05-01 08:53:56.292: V/PackageManager(918):     labelRes=0x7f070004 nonLocalizedLabel=null icon=0x7f020089 banner=0x0

05-01 08:53:56.292: V/PackageManager(918):     className=com.uc.browser.UCMobileApp

05-01 08:53:56.292: V/PackageManager(918):     processName=com.UCMobile

05-01 08:53:56.292: V/PackageManager(918):     taskAffinity=com.UCMobile

05-01 08:53:56.292: V/PackageManager(918):     uid=10073 flags=0x883e44 theme=0x0 flagsEx=0x0

05-01 08:53:56.292: V/PackageManager(918):     requiresSmallestWidthDp=0 compatibleWidthLimitDp=0 largestWidthLimitDp=0

05-01 08:53:56.292: V/PackageManager(918):     sourceDir=/data/app/com.UCMobile-1/base.apk

05-01 08:53:56.292: V/PackageManager(918):     seinfo=default

05-01 08:53:56.292: V/PackageManager(918):     dataDir=/data/data/com.UCMobile

05-01 08:53:56.292: V/PackageManager(918):     enabled=true targetSdkVersion=23 versionCode=648

05-01 08:53:56.292: V/PackageManager(918):     manageSpaceActivityName=com.uc.application.cleaner.SystemManageData.ManageDataActivity

05-01 08:53:56.292: V/PackageManager(918):     supportsRtl=false

05-01 08:53:56.292: D/PackageManager(918): ===== pa.mPref.Set =[Ljava.lang.String;@1053e89d, always=true

05-01 08:53:56.292: D/PackageManager(918): set[]=com.UCMobile

05-01 08:53:56.292: D/PackageManager(918): set[]=com.android.browser

05-01 08:53:56.292: D/PackageManager(918): ===== pa.mPref.Set =[Ljava.lang.String;@1053e89d

05-01 08:53:56.292: D/PackageManager(918): set[]=com.UCMobile

05-01 08:53:56.293: D/PackageManager(918): set[]=com.android.browser

05-01 08:53:56.293: I/PackageManager(918): Result set changed, dropping preferred activity for Intent { act=android.intent.action.VIEW dat=[http://wap.uc.cn](http://wap.uc.cn/) flg=0x8 } type null

05-01 08:53:56.293: V/PackageManager(918): Removing preferred activity since set changed ComponentInfo{com.UCMobile/com.UCMobile.main.UCMobile}

05-01 08:53:56.293: V/IntentResolver(918): Removing filter: PreferredActivity{0x43bb27d com.UCMobile/.main.UCMobile}

05-01 08:53:56.293: V/IntentResolver(918):       Action: "android.intent.action.VIEW"

05-01 08:53:56.293: V/IntentResolver(918):       Category: "android.intent.category.DEFAULT"

05-01 08:53:56.293: V/IntentResolver(918):       Scheme: "http"

05-01 08:53:56.293: V/IntentResolver(918):     Cleaning Lookup Maps:

05-01 08:53:56.293: V/IntentResolver(918):       Scheme: http

05-01 08:53:56.293: V/IntentResolver(918): Adding filter: PreferredActivity{0xb081112 com.UCMobile/.main.UCMobile}

05-01 08:53:56.293: V/IntentResolver(918):       Action: "android.intent.action.VIEW"

05-01 08:53:56.293: V/IntentResolver(918):       Category: "android.intent.category.DEFAULT"

05-01 08:53:56.293: V/IntentResolver(918):       Scheme: "http"

05-01 08:53:56.293: V/IntentResolver(918):     Building Lookup Maps:

05-01 08:53:56.293: V/IntentResolver(918):       Scheme: http

05-01 08:53:56.293: V/PackageManager(918): Preferred activity bookkeeping changed; writing restrictions