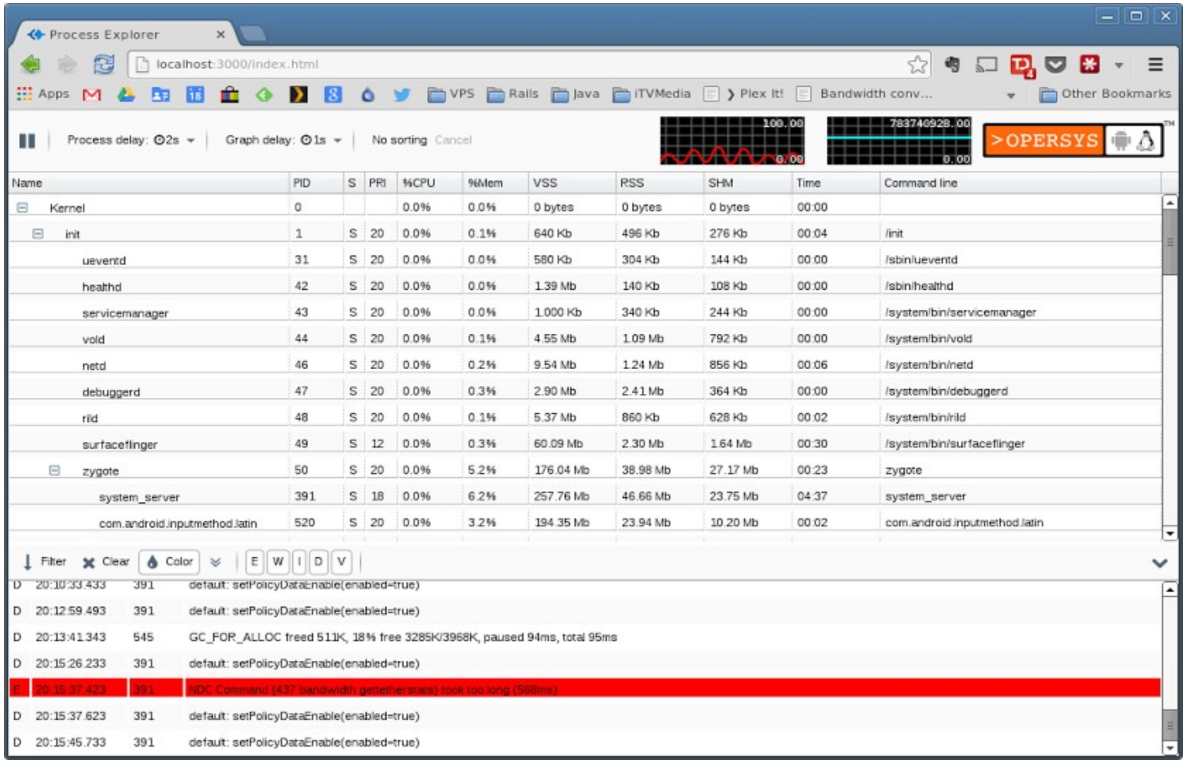
Developing Android Platform Tools

Android Builders Summit 2015

François-Denis Gonthier
@fdgonthier

About

- Author and maintainer of:



Name	PID	S	PRI	%CPU	%Mem	VSS	RSS	SHM	Time	Command line
⊟ Kernel	0			0.0%	0.0%	0 bytes	0 bytes	0 bytes	00:00	
⊟ init	1	S	20	0.0%	0.1%	640 Kb	496 Kb	276 Kb	00:04	/init
ueventd	31	S	20	0.0%	0.0%	580 Kb	304 Kb	144 Kb	00:00	/sbin/ueventd
healthd	42	S	20	0.0%	0.0%	1.39 Mb	140 Kb	108 Kb	00:00	/sbin/healthd
servicemanager	43	S	20	0.0%	0.0%	1.000 Kb	340 Kb	244 Kb	00:00	/system/bin/servicemanager
vold	44	S	20	0.0%	0.1%	4.55 Mb	1.09 Mb	792 Kb	00:00	/system/bin/vold
netd	46	S	20	0.0%	0.2%	9.54 Mb	1.24 Mb	856 Kb	00:06	/system/bin/netd
debuggerd	47	S	20	0.0%	0.3%	2.90 Mb	2.41 Mb	364 Kb	00:00	/system/bin/debuggerd
rild	48	S	20	0.0%	0.1%	5.37 Mb	860 Kb	628 Kb	00:02	/system/bin/rild
surfaceflinger	49	S	12	0.0%	0.3%	60.09 Mb	2.30 Mb	1.64 Mb	00:30	/system/bin/surfaceflinger
⊟ zygote	50	S	20	0.0%	5.2%	176.04 Mb	38.98 Mb	27.17 Mb	00:23	zygote
system_server	391	S	18	0.0%	6.2%	257.76 Mb	46.66 Mb	23.75 Mb	04:37	system_server
com.android.inputmethod.latin	520	S	20	0.0%	3.2%	194.35 Mb	23.94 Mb	10.20 Mb	00:02	com.android.inputmethod.latin

```
D   20:10:33.433   391   default: setPolicyDataEnable(enabled=true)
D   20:12:59.493   391   default: setPolicyDataEnable(enabled=true)
D   20:13:41.343   545   GC_FOR_ALLOC freed 511K, 18% free 3285K/3968K, paused 94ms, total 95ms
D   20:15:26.233   391   default: setPolicyDataEnable(enabled=true)
E   20:15:37.423   391   NDC Command {437 bandwidth gettetherstats} took too long (568ms)
D   20:15:37.623   391   default: setPolicyDataEnable(enabled=true)
D   20:15:45.733   391   default: setPolicyDataEnable(enabled=true)
```

- github.com/opersys/raidl

Agenda

- What's out there?
- What's the problem?
- Our Objectives
- Initial Set of Pain Points
- raidl – AIDL file lookup
- Architecture for Creating Monitoring Tools
- Process Explorer
- File Explorer
- Binder Explorer
- The Road Ahead

1. What's Out There Now?

- Official App Dev Tools
- Official Platform Dev Tools
- 3rd Party App Dev Tools
- 3rd Party Platform Dev Tools

1.1. Official App Dev tools

- Eclipse Android Development Kit
- Android Studio (IntelliJ)
- DDMS
- Plenty of documentation

1.2. Official Platform Dev Tools

- Tools on the previous pages
- gdb / gdbserver
- ftrace, systrace, atrace
- perf

1.3. 3rd Party App Dev Tools

- CrossWalk / Cordova
- Delphi
- Xamarin.Android
- etc.

1.4. 3rd Party Platform Dev Tools

- Qualcomm tools
- Intel tools, Nvidia tools, etc
- ARM Tools – DS-5
- JTAG -- Lauterbach

2. What's The Problem?

- Google obviously catering to app developers
  - App dev tools have nice finish
  - Platform dev tools ... 😵
- Official tools heavily tied to app dev IDE
  - Requires IDE-specific knowledge to extend/customize
  - Assumes official IDE is being used and/or is present
- Platform is huge

2. What's The Problem

- Documentation is often spartan
- Existing platform dev tools assume internals understanding
  - Do you truly know how to use “dumpsys procstats”, “dumpsys meminfo” or “vdc”
- Most platform tools can only be used on the command line
- Most 3rd party tools assume on-screen rendering of information

3. Our Objectives

- Reduce barrier to entry for platform development
- Catter for unmet patform developer needs
- Easy to use platform dev tools
- Build on lightweight/mainstream technologies:
  - No IDE-specific tie-in
  - Extensible language
  - Large ecosystem of reusable packages/add-ons
- Avoid monopolizing device screen

4. Initial Set of Pain Points

- Looking up AIDL interfaces
- Monitoring Processes
- Viewing / Manipulating the Filesystem
- Understanding Binder Relationships

4.1. Getting AIDL files

- find -name “*File.aidl”
- godir
- Android documentation
  - Focused on app developers
  - Doesn't cover everything

4.2. Process Monitoring

- ps / top
- htop (Cyanogenmod)
- Studio/Eclipse/DDMS/Monitor integrated
- On the Play Store...
  - ... hundreds of candidates
  - Few are aimed at developers

4.3. Filesystem Monitoring/Browsing

- ls, find
- adb push/pull
- On the Play Store...
  - ... hundreds of candidates
  - Few are aimed at developers

4.4. Binder Relationships

5. Raidl - Features

- Returns the AIDL interface of a service
  - AIDL based service
  - Best effort for other services (Activity service)
  - No interface for C++ service
  - No parameters names 😭

5.1. Example Output

```
root@generic:/data/local/tmp # ./raidl iface -n power
// Service: power, Interface: android.os.IPowerManager
package android.os;

interface IPowerManager {
    void acquireWakeLock(IBinder p1, int n2, String s3, String s4, WorkSource p5, String 
s6); // 1
    void acquireWakeLockWithUid(IBinder p1, int n2, String s3, String s4, int n5); // 2
    void releaseWakeLock(IBinder p1, int n2); // 3
    void updateWakeLockUids(IBinder p1, int[] p2); // 4
    void powerHint(int n1, int n2); // 5
    void updateWakeLockWorkSource(IBinder p1, WorkSource p2, String s3); // 6
    boolean isWakeLockLevelSupported(int n1); // 7
    void userActivity(long n1, int n2, int n3); // 8
    void wakeUp(long n1); // 9
    void goToSleep(long n1, int n2, int n3); // 10
    void nap(long n1); // 11
    boolean isInteractive(); // 12
    boolean isPowerSaveMode(); // 13
    boolean setPowerSaveMode(boolean p1); // 14
    void reboot(boolean p1, String s2, boolean p3); // 15
    void shutdown(boolean p1, boolean p2); // 16
    void crash(String s1); // 17
    void setStayOnSetting(int n1); // 18
    void setMaximumScreenOffTimeoutFromDeviceAdmin(int n1); // 19
    void setTemporaryScreenBrightnessSettingOverride(int n1); // 20
    void setTemporaryScreenAutoBrightnessAdjustmentSettingOverride(float p1); // 21
    void setAttentionLight(boolean p1, int n2); // 22
}
```

5.2. Raidl - Demo

5.3. Raidl – How Does It Work?

```
ServiceStubClass = Raidl.class.getClassLoader()
                              .loadClass(serviceClass.getCanonicalName()+"$Stub");

for (Field serviceField : serviceStubClass.getDeclaredFields()) {
    // Get the fields that look like transaction code.
}

for (Method serviceMethod : serviceClass.getMethods())
    serviceMethods.put(serviceMethod.getName(), serviceMethod);

for (Integer serviceCode : serviceCodesMethods.keySet()) {
    // ...

    if (serviceMethod != null && isRemoteMethod(serviceMethod))
        aidlService.addMethod(serviceCode, serviceMethod);
}
```

5.4. Raild - Integrate in AOSP Build

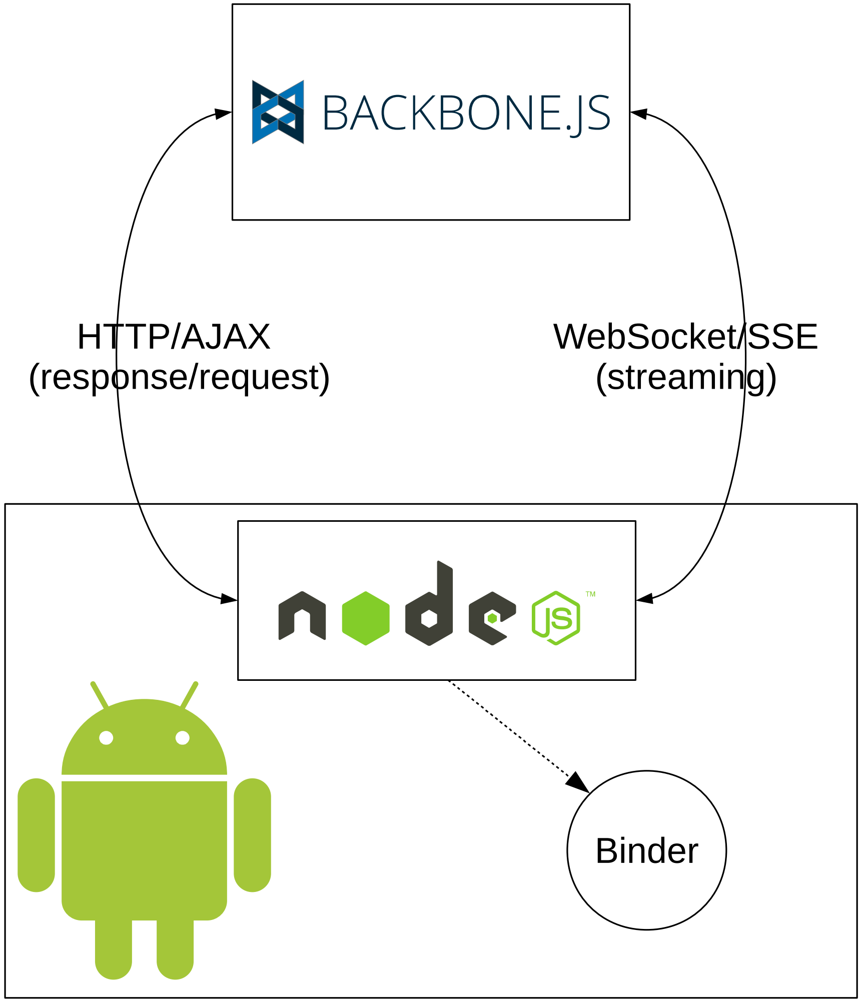
```
LOCAL_PATH:= $(call my-dir)
include $(CLEAR_VARS)

LOCAL_SRC_FILES := $(call all-java-files-under, src)
LOCAL_PACKAGE_NAME := raidl
LOCAL_MODULE_TAGS := optional
LOCAL_PROGUARD_ENABLED := disabled

include $(BUILD_PACKAGE)

include $(CLEAR_VARS)

LOCAL_SRC_FILES := raidl
LOCAL_MODULE_PATH := $(TARGET_OUT)/bin
LOCAL_MODULE_CLASS := EXECUTABLES
LOCAL_MODULE := raidl
LOCAL_MODULE_TAGS := optional
include $(BUILD_PREBUILT)
```

5.5. Raild - Running an .apk

- .apk == .jar (with some DEX code)
- `DexClassLoader:`“A class loader that loads classes from .jar and .apk files [...]”
- Ergo:

```
export CLASSPATH=/system/app/raidl.apk:/system/app/raidl/raidl.apk
exec app_process /system/app com.opersys.raidl.Raidl "$@"
```

6. Architecture for Creating Monitoring Tools



6.1. Tool architecture

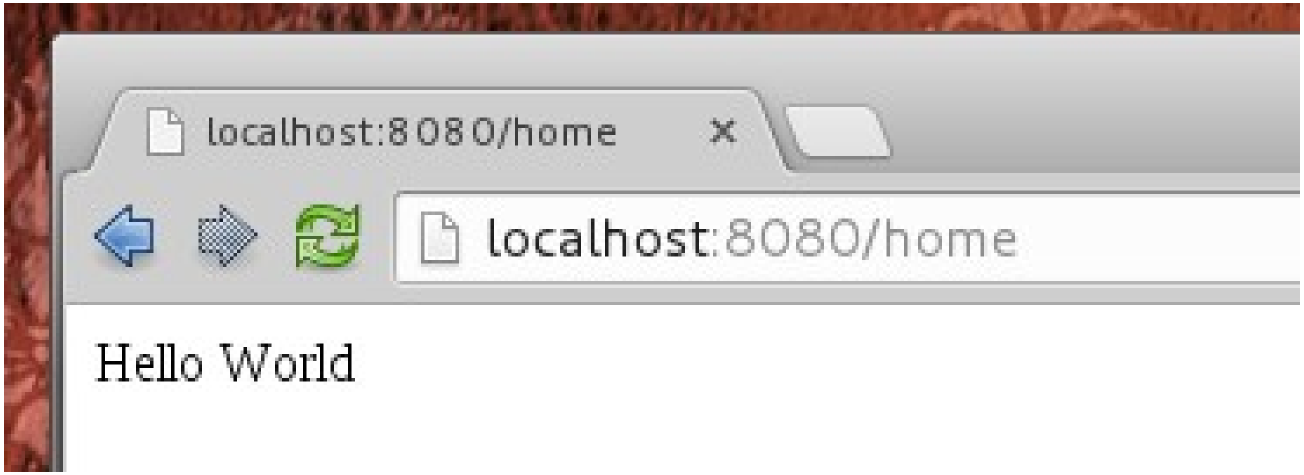
- Backend: Node.js + Express
- Frontend: Backbone.js + w2ui
- AJAX communication
- Websocket or Server-Sent events

6.2. Node.js in Android – Why?

- One language to rule them all: Javascript
- 132 510 Node.js packages
- Ease of use
- Web 2.0 support (SSE, WebSockets)
- Speed
  - V8
  - Binary modules
- **Runtime independence**
- Few actual alternatives: Go, C/C++, Python, etc.

6.3. Node.js – It's easy!

```
var express = require('express');
var app = express();

app.get('/home', function(req, res) {
 res.send('Hello World');
});

app.listen(process.env.PORT || 8080);
```



Hello World

6.4. Node.js in Android – Why not?

- Still pretty slow
- Runtime independence
  - Node is within its Linux bottle
- Difficult to package in Android
- It's Javascript
  - WAT! https://www.destroyallsoftware.com/talks/wat

6.5. How to use Node.js on Android

- Older versions (0.8), binaries available
  - Too old for comfort
- Development version (0.11, now 0.12) was patched with Android support
- Backported to 0.10
- V0.10 Binaries are avaible!
- Io.js and Node v0.12: TBD.
- https://github.com/fdgonthier/node

6.6. Distribution

- Extracted in local files
- Multiple binary packages
  - ARM, ARM + PIE, ia32, ia32 + PIE
- Started by a simple Android application
- Able to start as root

7. Process Explorer

- Browser based process manager
- Displays logcat (live!)
- Process statistics
  - /proc based
- Needs root access for better function
- Works on Chrome and Firefox

7. Process Explorer - Demo

8. File Explorer

- Browser based file manager for Android systems
- File upload/download
- File updates (live!)
- Needs root access for better function

8. File Explorer - Demo

9. Binder Explorer

- In development...
- Analysis of the links between Binder Services and Android applications
- Uses data from `/sys/kernel/debug/binder`
- Pushing further: JSLibBinder

9. Binder Explorer - Demo

9.1. Reaching Inside Android

- JSLibBinder – libbinder for Android

```
var Binder = require("jslibbinder"),
var sm = new Binder.ServiceManager();
var services = sm.list();
var i = 0;

console.log("Found " + services.length + " services:");
services.forEach(function (s) {
    console.log((i++) + "\t" + s
                + ": [" + sm.getService(s).getInterface() + "]");
});
```

10. The Road Ahead

- New Features:
  - Raidl – service to JS Binder interfaces
  - Process Explorer new version has:
    - More /proc data: memory, network, process maps, etc.
  - File Explorer ...
  - Binder explorer:
    - Allow JS calls to Binder
- New Tools:
  - We've got our ideas ;D
  - We'd like to hear from you: What are you looking for?