Android for .NET Developers Series
Getting Started
Dalvik Debug Monitor Server (DDMS)

Jim Wilson

jimw@jwhh.com

@hedgehogjim

http://facebook.com/hedgehogjim

Outline

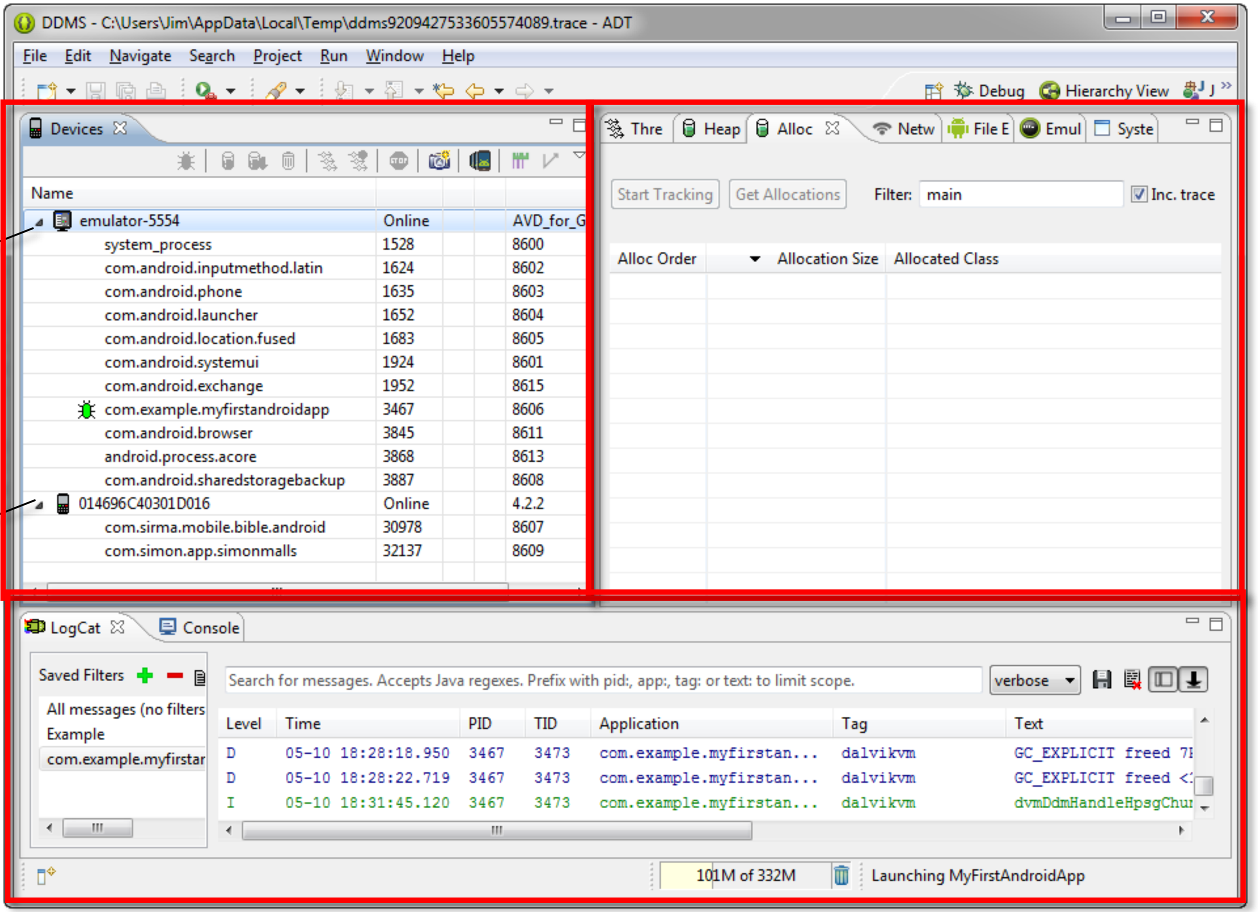
- What is DDMS?
- DDMS process visibility
- DDMS architecture
- DDMS process-oriented tools
- DDMS device/AVD-oriented tools

What is DDMS?

- DDMS is a tool that provides a wide variety of debugging features
  - List of connected devices and AVDs
    - Processes running on those devices/AVDs
  - Tools tabs
  - Logcat View

AVD

Device


DDMS - C:\Users\Jim\AppData\Local\Temp\ddms9209427533605574089.trace - ADT
File Edit Navigate Search Project Run Window Help
Debug
Hierarchy View
Devices
Thre Heap Alloc Netw File E Emul Syste
Start Tracking
Get Allocations
LogCat
Console
Saved Filters

Name		
emulator-5554	Online	AVD_for_G
system_process	1528	8600
com.android.inputmethod.latin	1624	8602
com.android.phone	1635	8603
com.android.launcher	1652	8604
com.android.location.fused	1683	8605
com.android.systemui	1924	8601
com.android.exchange	1952	8615
com.example.myfirstandroidapp	3467	8606
com.android.browser	3845	8611
android.process.acore	3868	8613
com.android.sharedstoragebackup	3887	8608
014696C40301D016	Online	4.2.2
com.sirma.mobile.bible.android	30978	8607
com.simon.app.simonmalls	32137	8609

Filter: main ✓ Inc. trace

Alloc Order	Allocation Size	Allocated Class

All messages (no filters
Example
com.example.myfirstar

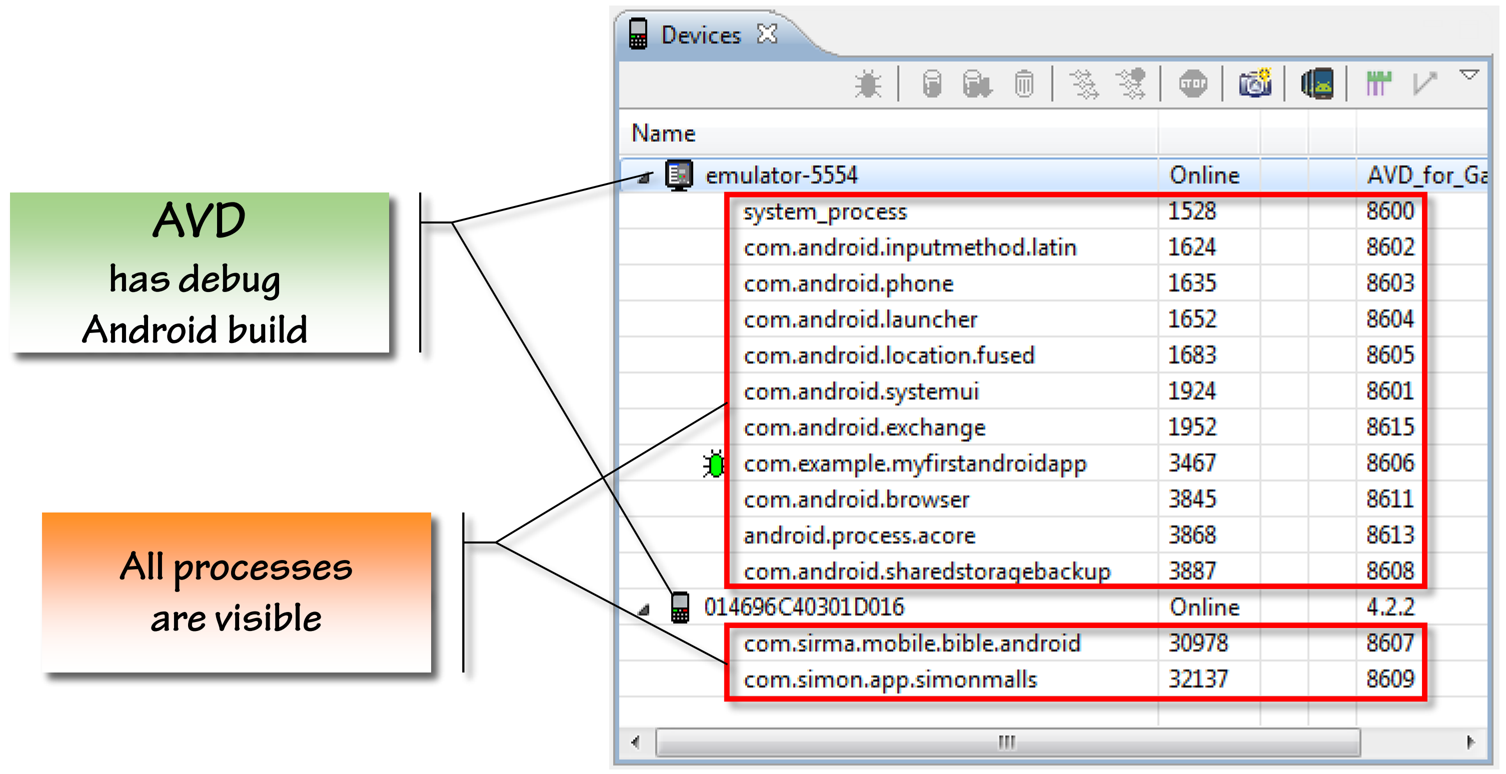
Search for messages. Accepts Java regexes. Prefix with pid:, app:, tag: or text: to limit scope. verbose

```
Level  Time                PID   TID   Application              Tag       Text
D      05-10 18:28:18.950  3467  3473  com.example.myfirstan... dalvikvm  GC_EXPLICIT freed 7I
D      05-10 18:28:22.719  3467  3473  com.example.myfirstan... dalvikvm  GC_EXPLICIT freed <
I      05-10 18:31:45.120  3467  3473  com.example.myfirstan... dalvikvm  dvmDdmHandleHpsgChu
```

101M of 332M   Launching MyFirstAndroidApp

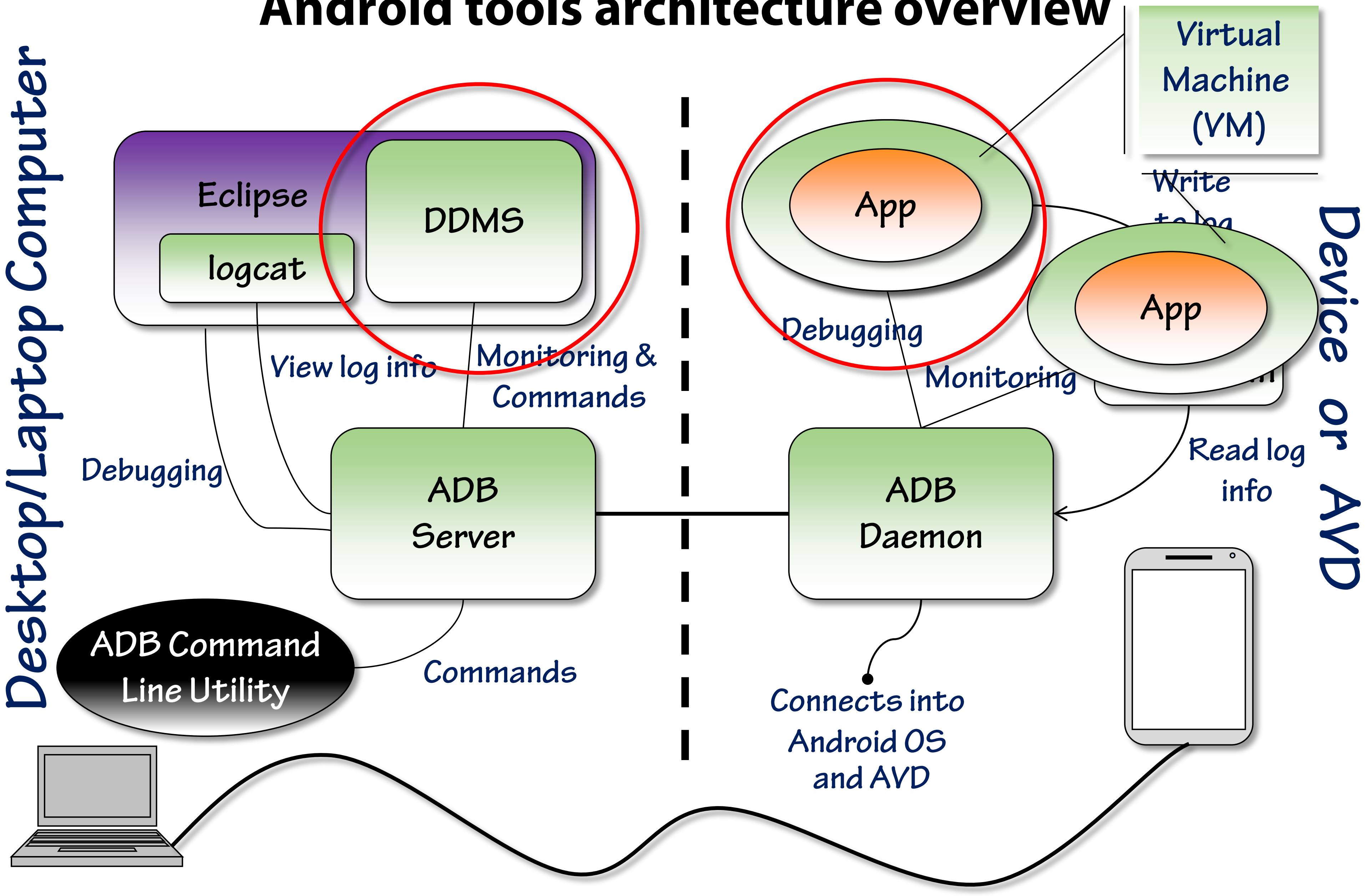
DDMS process visibility

- **For processes to be visible, debugging must be enabled**
  - Most Android devices uses a standard (non-debug) Android OS build
    - Only those processes marked as debuggable are visible
  - AVDs normally run a debug Android OS build
    - All processes are visible

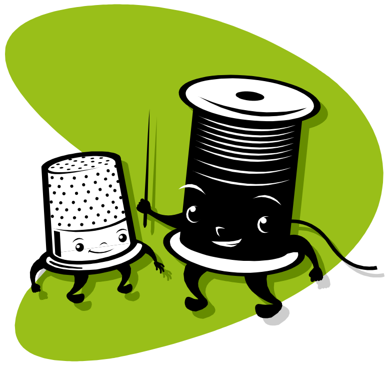


Name		
emulator-5554	Online	AVD_for_Ga
system_process	1528	8600
com.android.inputmethod.latin	1624	8602
com.android.phone	1635	8603
com.android.launcher	1652	8604
com.android.location.fused	1683	8605
com.android.systemui	1924	8601
com.android.exchange	1952	8615
com.example.myfirstandroidapp	3467	8606
com.android.browser	3845	8611
android.process.acore	3868	8613
com.android.sharedstoragebackup	3887	8608
014696C40301D016	Online	4.2.2
com.sirma.mobile.bible.android	30978	8607
com.simon.app.simonmalls	32137	8609

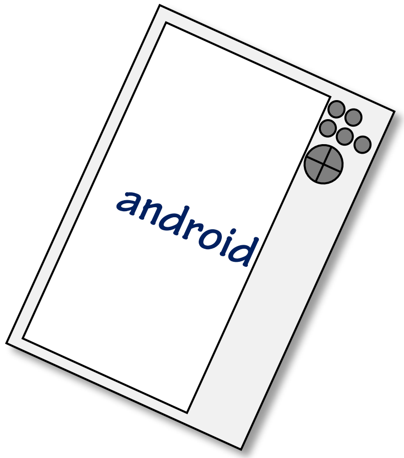
Android tools architecture overview



DDMS tools

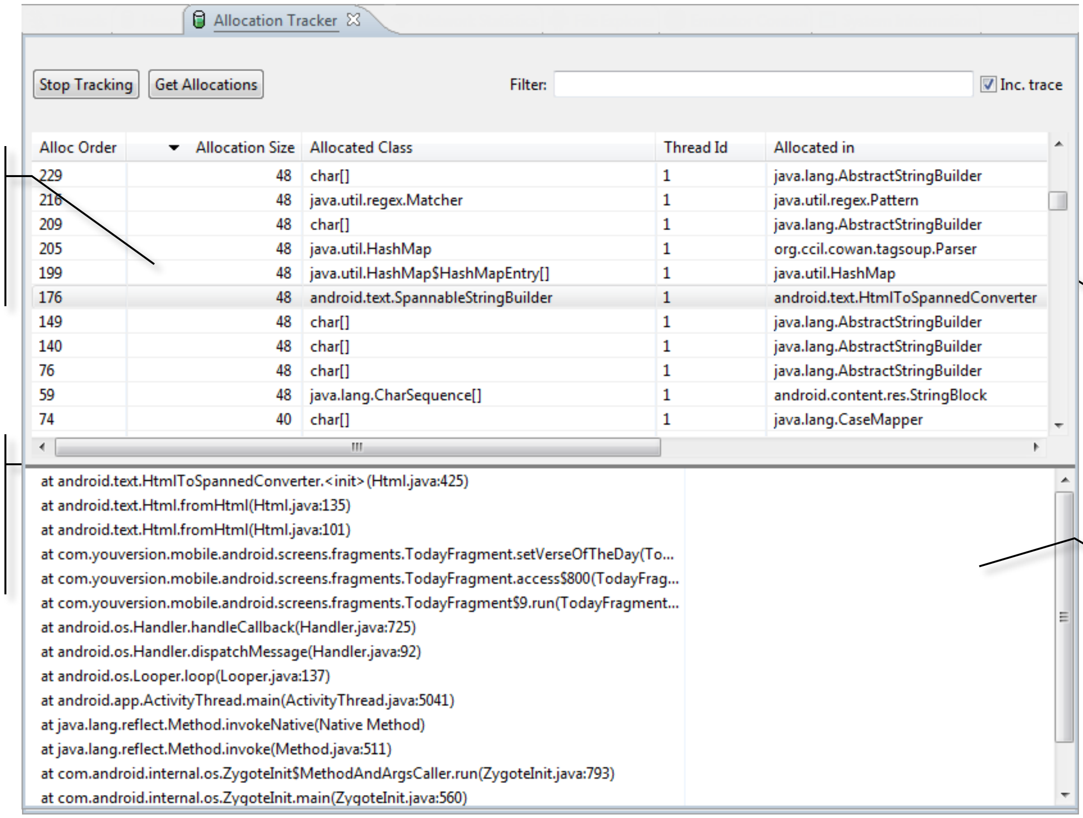


- DDMS incorporates many tools
  - Process-oriented
    - Memory usage
    - Thread activity
    - Network activity
  - Device/AVD-oriented
    - Access device/AVD file system
    - Screen capture
    - AVD control
  - Even more
    - For more information: http://bit.ly/12mdd5P



Memory usage

- **DDMS provides two primary memory usage tools**
  - Heap viewer
    - Provides a high-level view of app's general memory usage
  - Allocation Tracker
    - Detailed object by object memory allocation information



Alloc Order	Allocation Size	Allocated Class	Thread Id	Allocated in
229	48	char[]	1	java.lang.AbstractStringBuilder
216	48	java.util.regex.Matcher	1	java.util.regex.Pattern
209	48	char[]	1	java.lang.AbstractStringBuilder
205	48	java.util.HashMap	1	org.ccil.cowan.tagsoup.Parser
199	48	java.util.HashMap$HashMapEntry[]	1	java.util.HashMap
176	48	android.text.SpannableStringBuilder	1	android.text.HtmlToSpannedConverter
149	48	char[]	1	java.lang.AbstractStringBuilder
140	48	char[]	1	java.lang.AbstractStringBuilder
76	48	char[]	1	java.lang.AbstractStringBuilder
59	48	java.lang.CharSequence[]	1	android.content.res.StringBlock
74	40	char[]	1	java.lang.CaseMapper

```
at android.text.HtmlToSpannedConverter.<init>(Html.java:425)
at android.text.Html.fromHtml(Html.java:135)
at android.text.Html.fromHtml(Html.java:101)
at com.youversion.mobile.android.screens.fragments.TodayFragment.setVerseOfTheDay(To...
at com.youversion.mobile.android.screens.fragments.TodayFragment.access$800(TodayFrag...
at com.youversion.mobile.android.screens.fragments.TodayFragment$9.run(TodayFragment...
at android.os.Handler.handleCallback(Handler.java:725)
at android.os.Handler.dispatchMessage(Handler.java:92)
at android.os.Looper.loop(Looper.java:137)
at android.app.ActivityThread.main(ActivityThread.java:5041)
at java.lang.reflect.Method.invokeNative(Native Method)
at java.lang.reflect.Method.invoke(Method.java:511)
at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:793)
at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:560)
```

List of data types allocated

Distribution of Selected memory

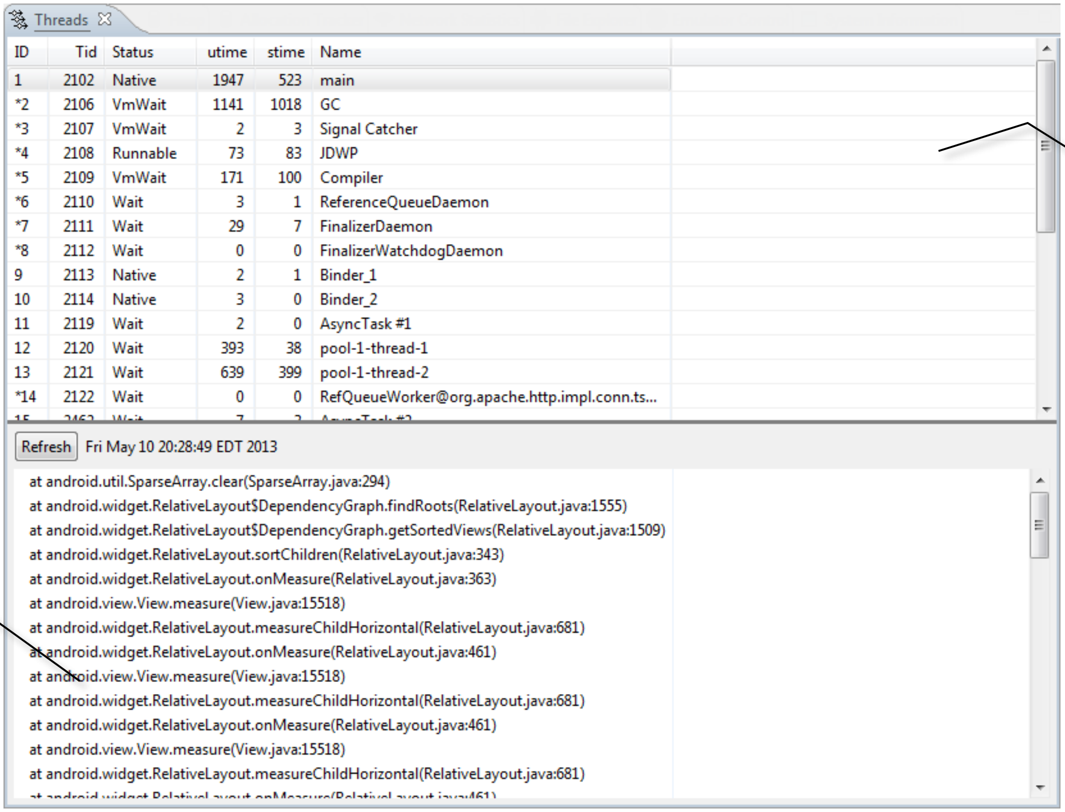
List of memory "buckets" in the managed heap

Call stack where selected allocation occurred

Threads

Threads viewer

- Shows threads running in app and each thread's current call stack



ID	Tid	Status	utime	stime	Name
1	2102	Native	1947	523	main
*2	2106	VmWait	1141	1018	GC
*3	2107	VmWait	2	3	Signal Catcher
*4	2108	Runnable	73	83	JDWP
*5	2109	VmWait	171	100	Compiler
*6	2110	Wait	3	1	ReferenceQueueDaemon
*7	2111	Wait	29	7	FinalizerDaemon
*8	2112	Wait	0	0	FinalizerWatchdogDaemon
9	2113	Native	2	1	Binder_1
10	2114	Native	3	0	Binder_2
11	2119	Wait	2	0	AsyncTask #1
12	2120	Wait	393	38	pool-1-thread-1
13	2121	Wait	639	399	pool-1-thread-2
*14	2122	Wait	0	0	RefQueueWorker@org.apache.http.impl.conn.ts...
15	2462	Wait	7	2	AsyncTask #2

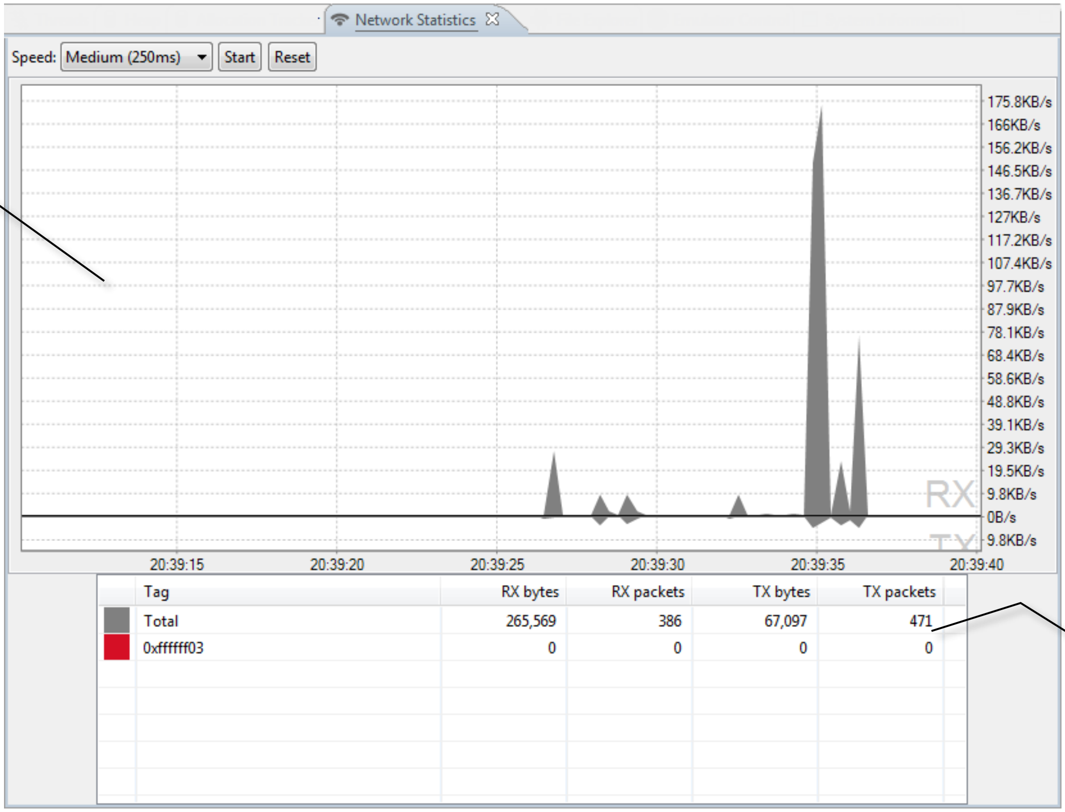
Fri May 10 20:28:49 EDT 2013

```
at android.util.SparseArray.clear(SparseArray.java:294)
at android.widget.RelativeLayout$DependencyGraph.findRoots(RelativeLayout.java:1555)
at android.widget.RelativeLayout$DependencyGraph.getSortedViews(RelativeLayout.java:1509)
at android.widget.RelativeLayout.sortChildren(RelativeLayout.java:343)
at android.widget.RelativeLayout.onMeasure(RelativeLayout.java:363)
at android.view.View.measure(View.java:15518)
at android.widget.RelativeLayout.measureChildHorizontal(RelativeLayout.java:681)
at android.widget.RelativeLayout.onMeasure(RelativeLayout.java:461)
at android.view.View.measure(View.java:15518)
at android.widget.RelativeLayout.measureChildHorizontal(RelativeLayout.java:681)
at android.widget.RelativeLayout.onMeasure(RelativeLayout.java:461)
at android.view.View.measure(View.java:15518)
at android.widget.RelativeLayout.measureChildHorizontal(RelativeLayout.java:681)
at android.widget.RelativeLayout.onMeasure(RelativeLayout.java:461)
```

Network activity

- **Network statistics**
  - Data sent and received by app

Data sent and received over time

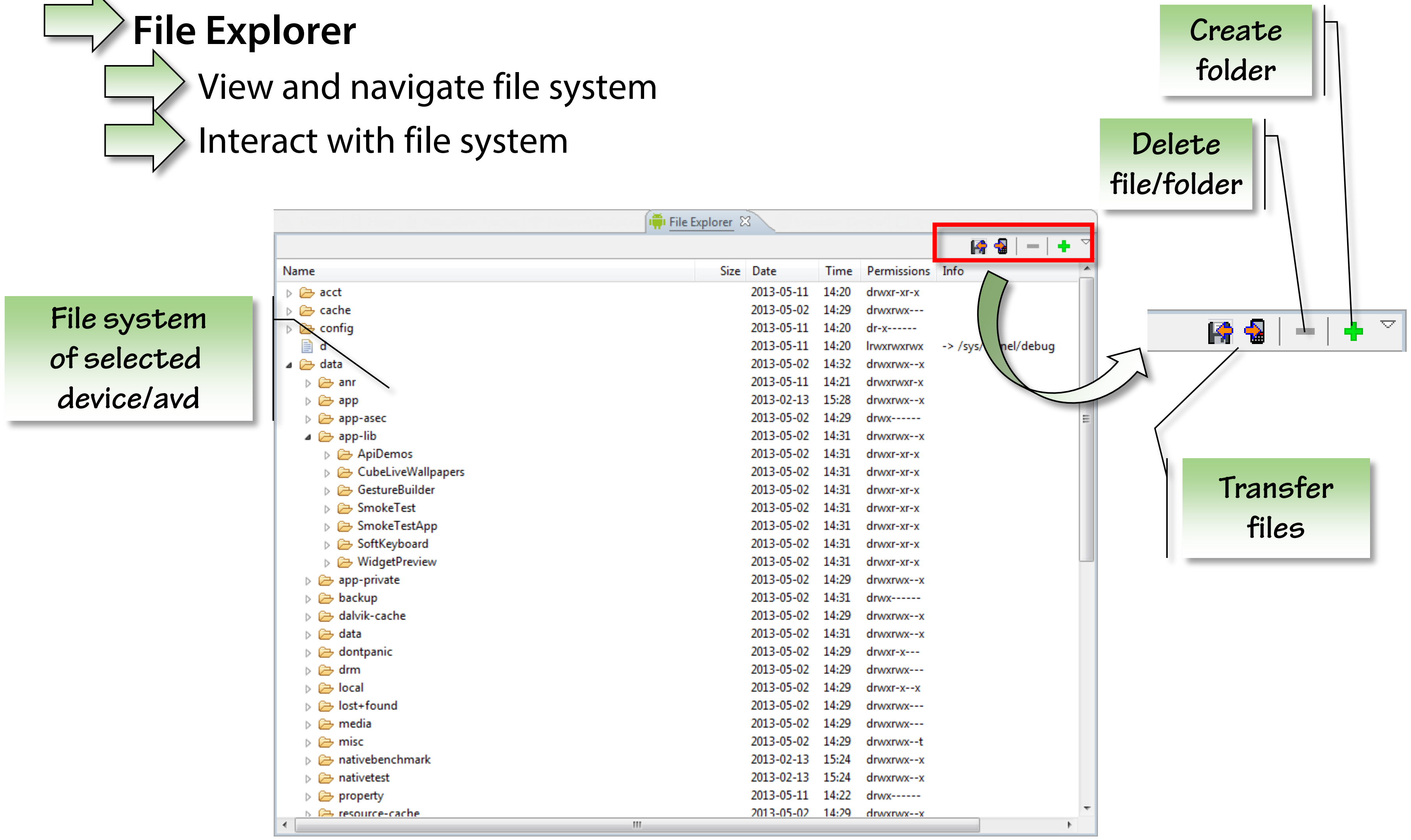


	Tag	RX bytes	RX packets	TX bytes	TX packets
⬛	Total	265,569	386	67,097	471
🟥	0xffffff03	0	0	0	0

Data sent and received totals

File system

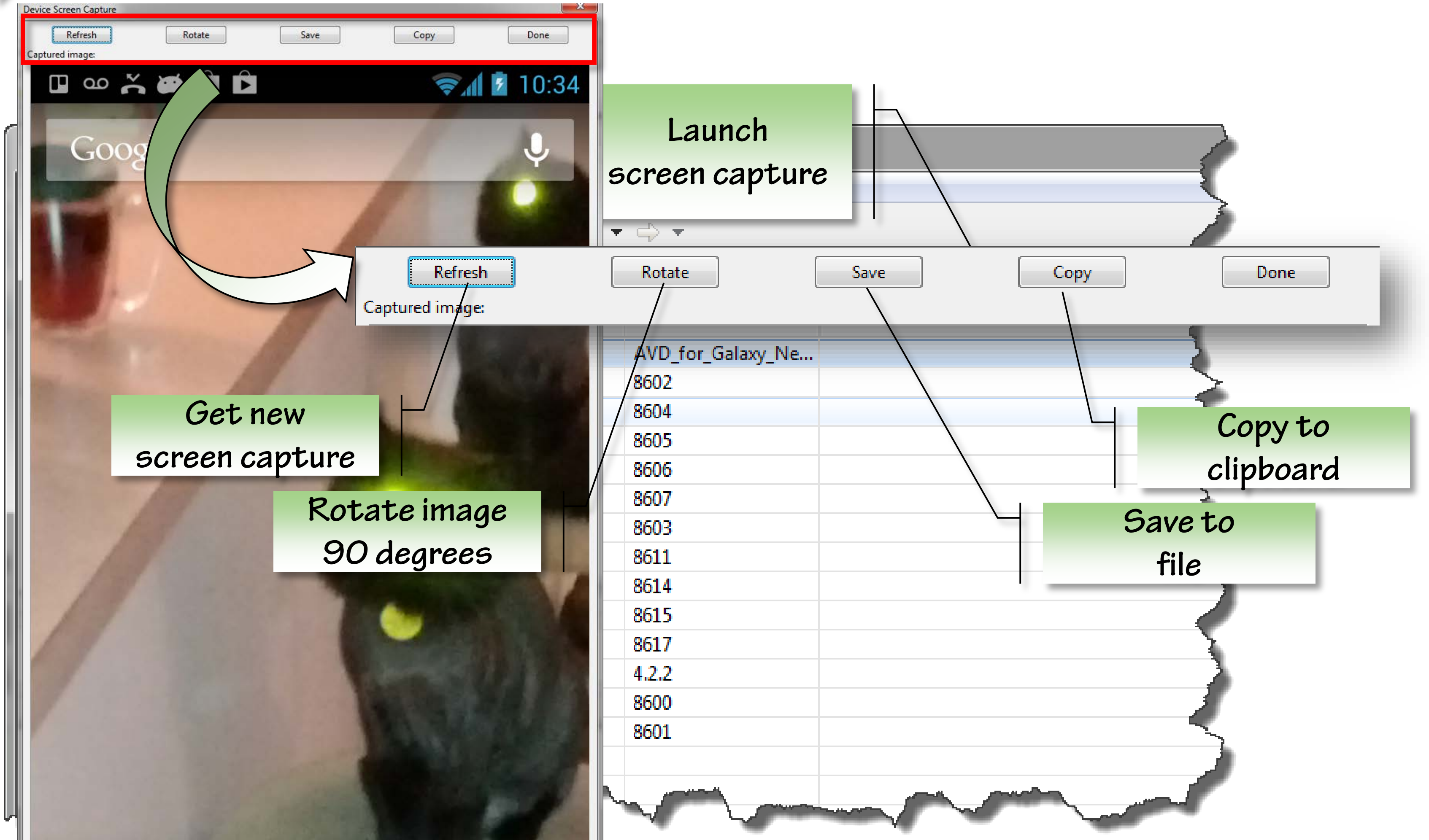
- **File Explorer**
  - View and navigate file system
  - Interact with file system



Name	Size	Date	Time	Permissions	Info
acct		2013-05-11	14:20	drwxr-xr-x	
cache		2013-05-02	14:29	drwxrwx---	
config		2013-05-11	14:20	dr-x------	
d		2013-05-11	14:20	lrwxrwxrwx	-> /sys/ nel/debug
data		2013-05-02	14:32	drwxrwx--x	
anr		2013-05-11	14:21	drwxrwxr-x	
app		2013-02-13	15:28	drwxrwx--x	
app-asec		2013-05-02	14:29	drwx------	
app-lib		2013-05-02	14:31	drwxrwx--x	
ApiDemos		2013-05-02	14:31	drwxr-xr-x	
CubeLiveWallpapers		2013-05-02	14:31	drwxr-xr-x	
GestureBuilder		2013-05-02	14:31	drwxr-xr-x	
SmokeTest		2013-05-02	14:31	drwxr-xr-x	
SmokeTestApp		2013-05-02	14:31	drwxr-xr-x	
SoftKeyboard		2013-05-02	14:31	drwxr-xr-x	
WidgetPreview		2013-05-02	14:31	drwxr-xr-x	
app-private		2013-05-02	14:29	drwxrwx--x	
backup		2013-05-02	14:31	drwx------	
dalvik-cache		2013-05-02	14:29	drwxrwx--x	
data		2013-05-02	14:31	drwxrwx--x	
dontpanic		2013-05-02	14:29	drwxr-x---	
drm		2013-05-02	14:29	drwxrwx---	
local		2013-05-02	14:29	drwxr-x--x	
lost+found		2013-05-02	14:29	drwxrwx---	
media		2013-05-02	14:29	drwxrwx---	
misc		2013-05-02	14:29	drwxrwx--t	
nativebenchmark		2013-02-13	15:24	drwxrwx--x	
nativetest		2013-02-13	15:24	drwxrwx--x	
property		2013-05-11	14:22	drwx------	
resource-cache		2013-05-02	14:29	drwxrwx--x	

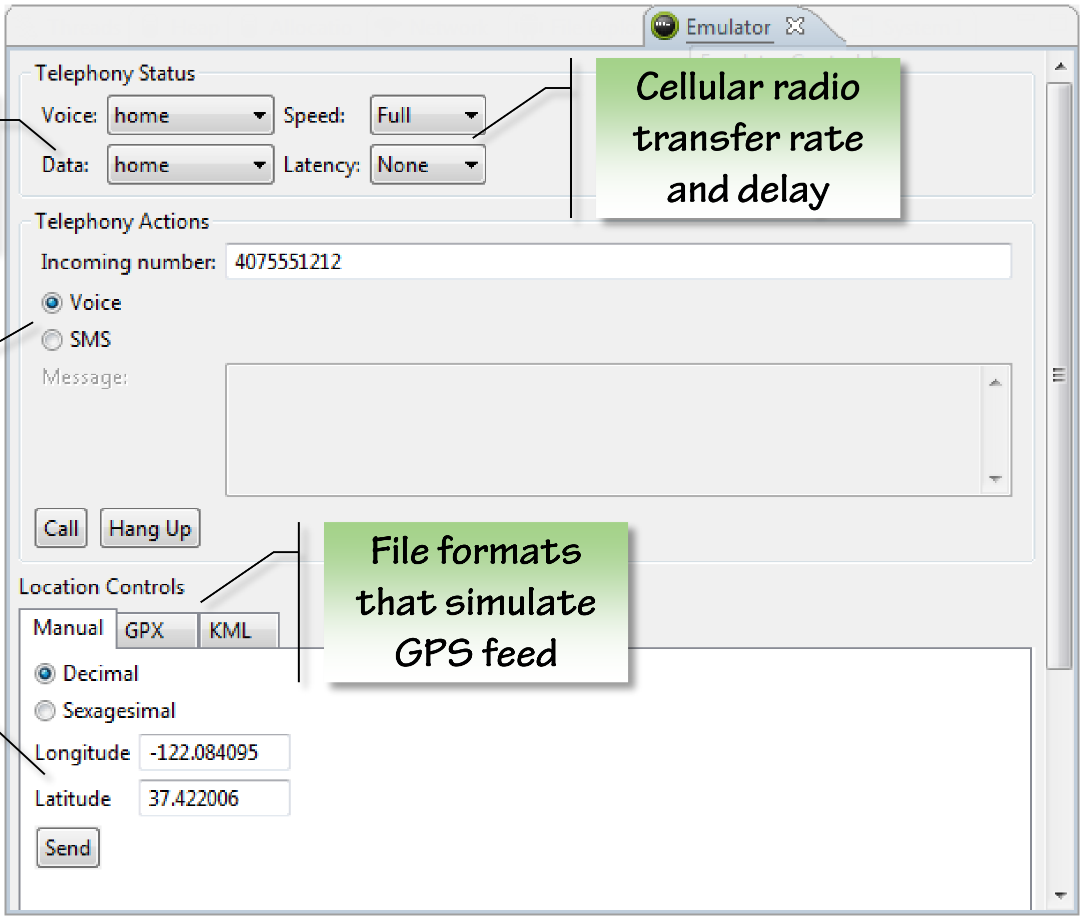
Screen shots

- **Screen capture**
  - Creates a static image of the selected device or AVD screen



Emulator

- **Emulator management**
  - Modify the emulator environment
  - Interact with the emulator



Emulator

Telephony Status

Voice: home — Speed: Full

Data: home — Latency: None

Telephony Actions

Incoming number: 4075551212

● Voice

○ SMS

Message:

Call — Hang Up

Location Controls

Manual | GPX | KML

● Decimal

○ Sexagesimal

Longitude: -122.084095

Latitude: 37.422006

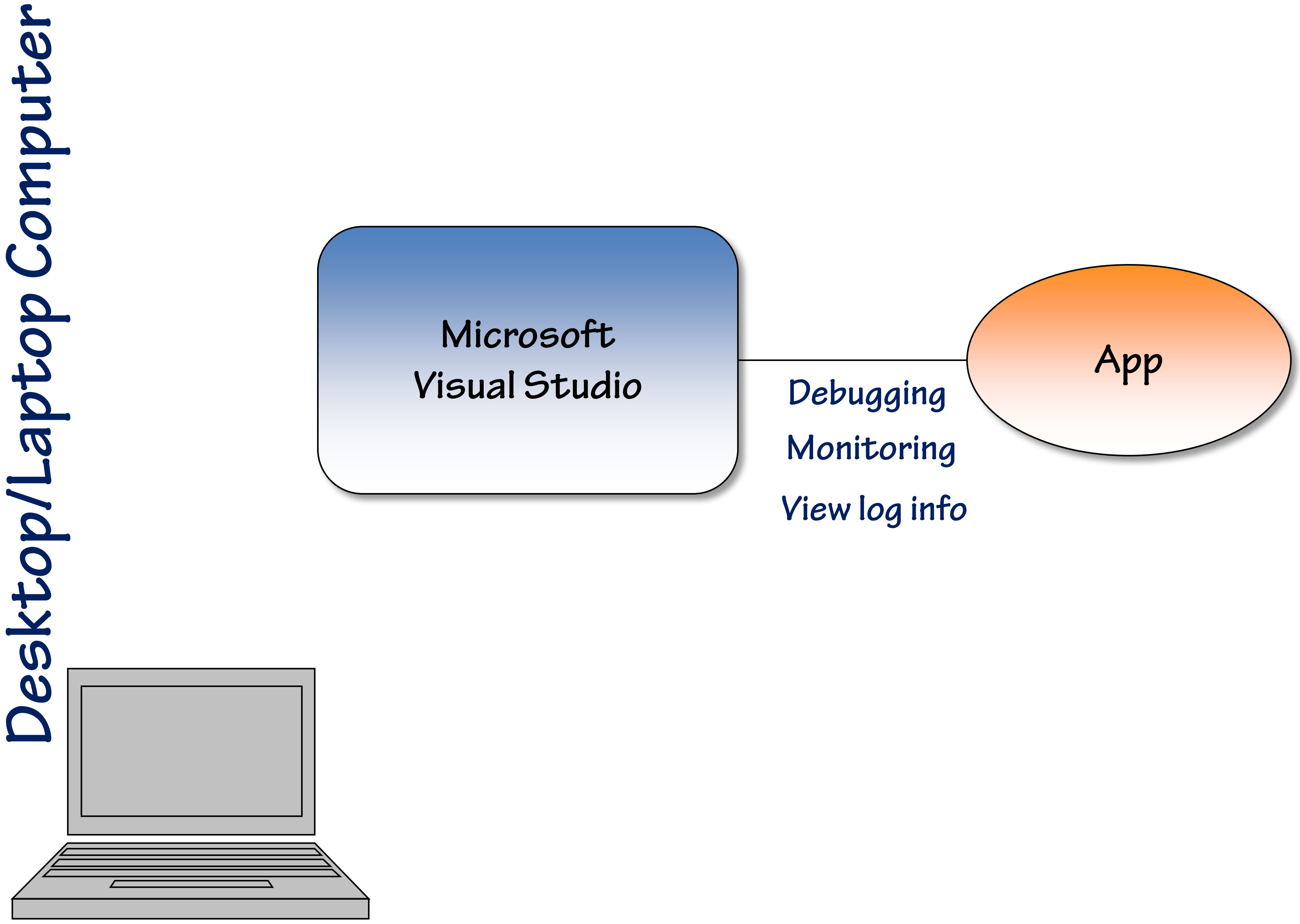
Send

Cellular radio behavior
Cellular radio transfer rate and delay
Simulate phone calls and text messaging
File formats that simulate GPS feed
Set the current GPS coordinates

Summary

- **DDMS is an essential debugging tool**
- **Debugging must be enabled for DDMS to access a process**
  - On most devices, limited to those processes explicitly set to debuggable
  - On most AVDs, Android is debug-build making all processes visible
- **DDMS provides rich process monitoring**
  - Memory
  - Threads
  - Network
- **DDMS deeply interacts with devices and emulators (AVDs)**
  - File system management
  - Screen capture
  - Emulator control

.NET tools architecture overview

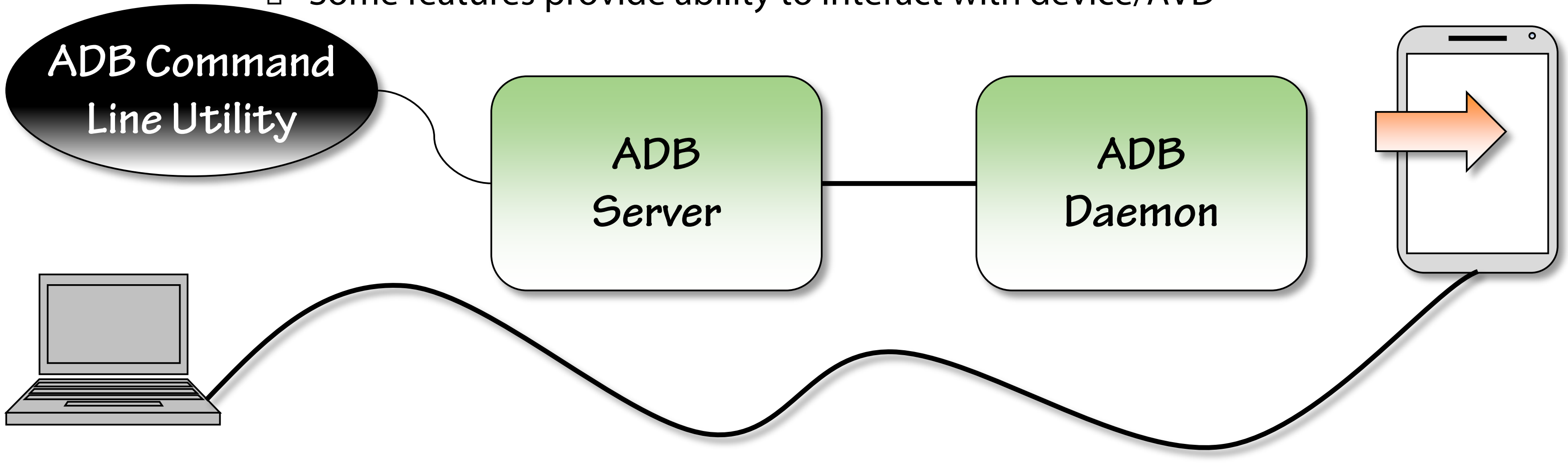


Key Android debugging tools

- **Majority of Android development handled with 4 tools**
  - Android Debug Bridge (ADB)
  - Eclipse
  - Logcat
  - Dalvik Debug Monitor Server (DDMS)
  - Many more tools available
    - Complete list at http://bit.ly/11Wd2PC

Android Debug Bridge (ADB)

- **Android Debug Bridge is the most important development tool**
  - Makes interaction between your desktop and device/AVD possible
    - Daemon running on the device
    - Server process running on your desktop
    - If ADB isn't working, no aspect of debugging works
  - Provides a command-line interface through the adb utility
    - Located in `<install-folder>\sdk\platform-tools`
      - Useful to add to your "path" environment variable
    - Some features provide control over ADB processes & device/AVD connections
    - Some features provide ability to interact with device/AVD



adb utility process control commands

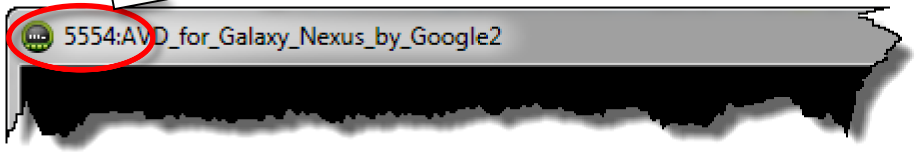
- **adb utility manages ADB processes and device/AVD connections**
  - One of the most important commands shows connected devices/AVDs
    - Run adb passing the ***devices*** command
    - Shows each device/AVD along with ADB assigned serial number
  - Controlling adb server lifetime
    - Use ***kill-server*** command to signal current ADB server instance to be shutdown
      - If works correctly, provides no feedback
    - ADB server will normally restart automatically when needed
    - Can assure that an instance is running with ***start-server*** command

Real Device

Emulator

```
> adb devices
List of devices attached
00000X99999X123       device
emulator-5554    device
```

```
> adb kill-server
```



adb utility device commands

- adb utility can perform actions on device/AVD
  - Provides a rich set of features and capabilities
    - Copy files between desktop & device/AVD
    - Can install/uninstall app on device/AVD
    - Much more – complete list at http://bit.ly/11WGTHF
  - Can open a interactive Linux shell on device
    - Use ***shell*** command with no arguments
  - Can issue a command-line that runs within a Linux shell on the device
    - Use ***shell*** command followed by Linux command-line
    - Returns immediately

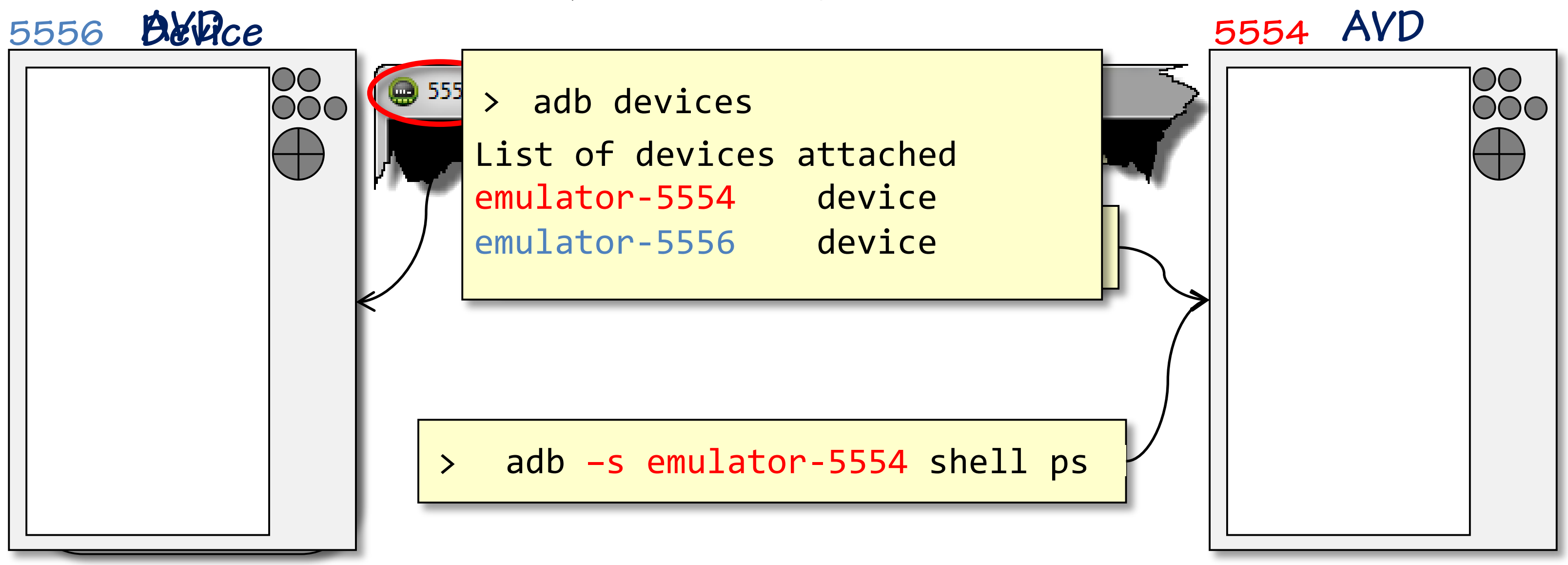
```
> adb shell
shell@android:/ $ cd ~
cd ~
shell@android:/data $  ps
ps
...   ... ...   ... ...   ...
...   ... ...   ... ...   ...
shell@android:/data $ exit
>
```

Running in a Linux shell on device/AVD

```
> adb shell ps
...   ... ...   ... ...   ...
...   ... ...   ... ...   ...
>
```

Identifying the adb utility target

- **Multiple connected devices/AVDs require special handling**
  - When multiple connections, you must specify the adb command target
    - When only a device or only an AVD, target could be inferred
  - Provides short-hand for common scenarios
    - To target the only connected device, include ***–d*** prior to the command
    - To target the only open AVD, include ***–e*** prior to the command
  - Multiple connections of the same type, require explicit target
    - Use ***–s*** followed by the ADB assigned serial number

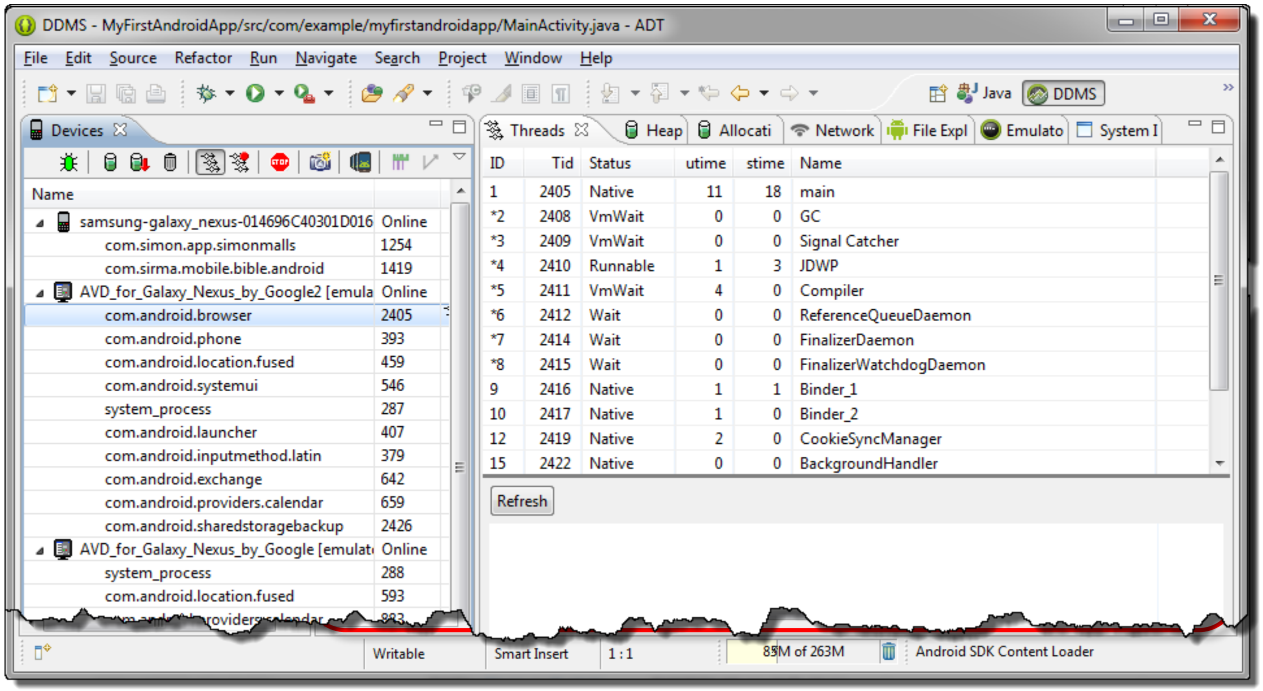


```
>  adb devices
List of devices attached
emulator-5554     device
emulator-5556     device
```

```
>  adb -s emulator-5554 shell ps
```

Eclipse

- **Eclipse is the hub of your development work**
  - Serves as the editor, source code debugger, project manager
  - It is truly the Integrated Development Environment
  - Serves as the host user interface for other Android tools
    - Logcat is available as a console-style View
    - DDMS is available as Perspective



DDMS - MyFirstAndroidApp/src/com/example/myfirstandroidapp/MainActivity.java - ADT

Name	
samsung-galaxy_nexus-014696C40301D016	Online
com.simon.app.simonmalls	1254
com.sirma.mobile.bible.android	1419
AVD_for_Galaxy_Nexus_by_Google2 [emula	Online
com.android.browser	2405
com.android.phone	393
com.android.location.fused	459
com.android.systemui	546
system_process	287
com.android.launcher	407
com.android.inputmethod.latin	379
com.android.exchange	642
com.android.providers.calendar	659
com.android.sharedstoragebackup	2426
AVD_for_Galaxy_Nexus_by_Google [emulat	Online
system_process	288
com.android.location.fused	593

ID	Tid	Status	utime	stime	Name
1	2405	Native	11	18	main
*2	2408	VmWait	0	0	GC
*3	2409	VmWait	0	0	Signal Catcher
*4	2410	Runnable	1	3	JDWP
*5	2411	VmWait	4	0	Compiler
*6	2412	Wait	0	0	ReferenceQueueDaemon
*7	2414	Wait	0	0	FinalizerDaemon
*8	2415	Wait	0	0	FinalizerWatchdogDaemon
9	2416	Native	1	1	Binder_1
10	2417	Native	1	0	Binder_2
12	2419	Native	2	0	CookieSyncManager
15	2422	Native	0	0	BackgroundHandler

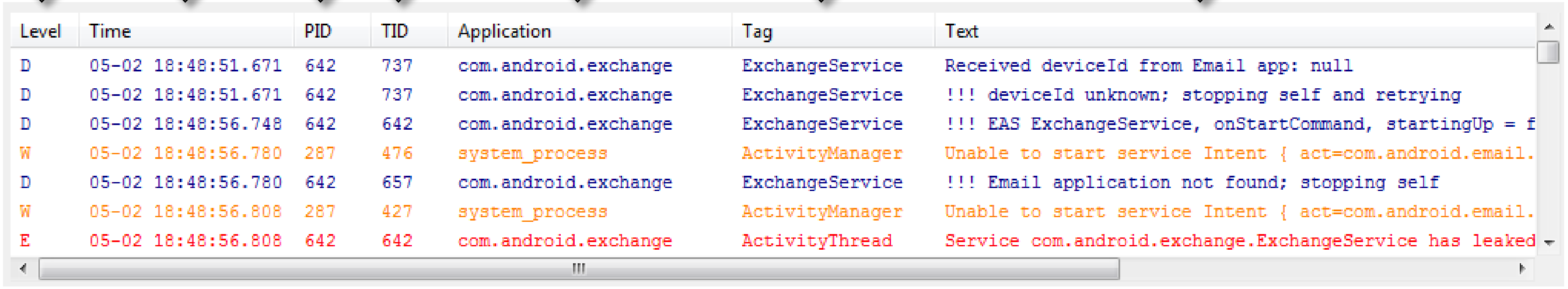
Logcat

- **Logcat is the Android logging system**
  - Serves as a message repository
    - Records information about system events
    - Apps can write messages
  - Messages written to logcat using android.util.Log class static methods
  - Can be easily read and filtered
    - Available via logcat view within Eclipse
    - Accessible through adb utility using ***logcat*** command

```
Log.e("MyApp", "Something's wrong");
```

Logcat structure

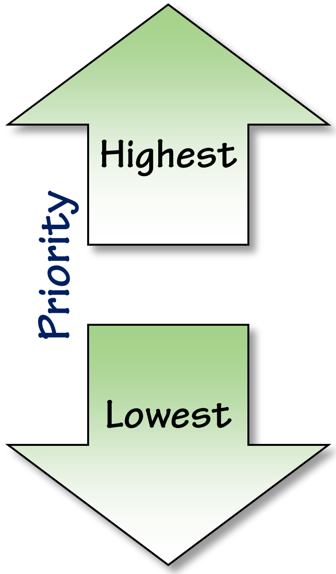
- Logcat information is stored as a consistent structured
  - Level
    - Indicates importance/severity
  - Time stamp
  - Process and thread id
  - Name of source Application
  - Tag
    - Application defined label
    - Intended to identify system, component or method
  - Text



Level	Time	PID	TID	Application	Tag	Text
D	05-02 18:48:51.671	642	737	com.android.exchange	ExchangeService	Received deviceId from Email app: null
D	05-02 18:48:51.671	642	737	com.android.exchange	ExchangeService	!!! deviceId unknown; stopping self and retrying
D	05-02 18:48:56.748	642	642	com.android.exchange	ExchangeService	!!! EAS ExchangeService, onStartCommand, startingUp = f
W	05-02 18:48:56.780	287	476	system_process	ActivityManager	Unable to start service Intent { act=com.android.email.
D	05-02 18:48:56.780	642	657	com.android.exchange	ExchangeService	!!! Email application not found; stopping self
W	05-02 18:48:56.808	287	427	system_process	ActivityManager	Unable to start service Intent { act=com.android.email.
E	05-02 18:48:56.808	642	642	com.android.exchange	ActivityThread	Service com.android.exchange.ExchangeService has leaked

Logcat levels

- **Levels are important to both creating a viewing logcat entries**
  - Indicate the relative importance of message
  - Managed as a hierarchy
    - Setting view filter to a level includes the level and all higher levels



Level	Label	Log method
Silent	FILTER ONLY	
Assert	A	`Log.wtf`
Error	E	`Log.e`
Warning	W	`Log.w`
Info	I	`Log.i`
Debug	D	`Log.d`
Verbose	V	`Log.v`

No messages Can be written at this level

What a Terrible Failure

Logcat buffers

- **Logcat segments messages into buffers**
  - Overwhelming majority of messages go into the “main” buffer
    - Standard Log class method calls write here
    - All messages viewed from Eclipse are read from the “main” buffer
  - Two special purpose buffers available
    - events: Shows all system events
    - radio: Shows activity of system radios (cellular, etc.)
  - Only accessible through adb command-line utility
    - Use **logcat** command with ***–b*** option followed by the buffer name

DDMS tools

- **DDMS incorporates many tools**
  - Process-oriented
    - Memory usage
    - Thread activity
    - Method profiling
    - Network activity
  - Device/AVD-oriented
    - Access device/AVD file system
    - Screen capture
    - AVD control
  - Even more
    - For more inform ly/12mdd5P