# Debugging

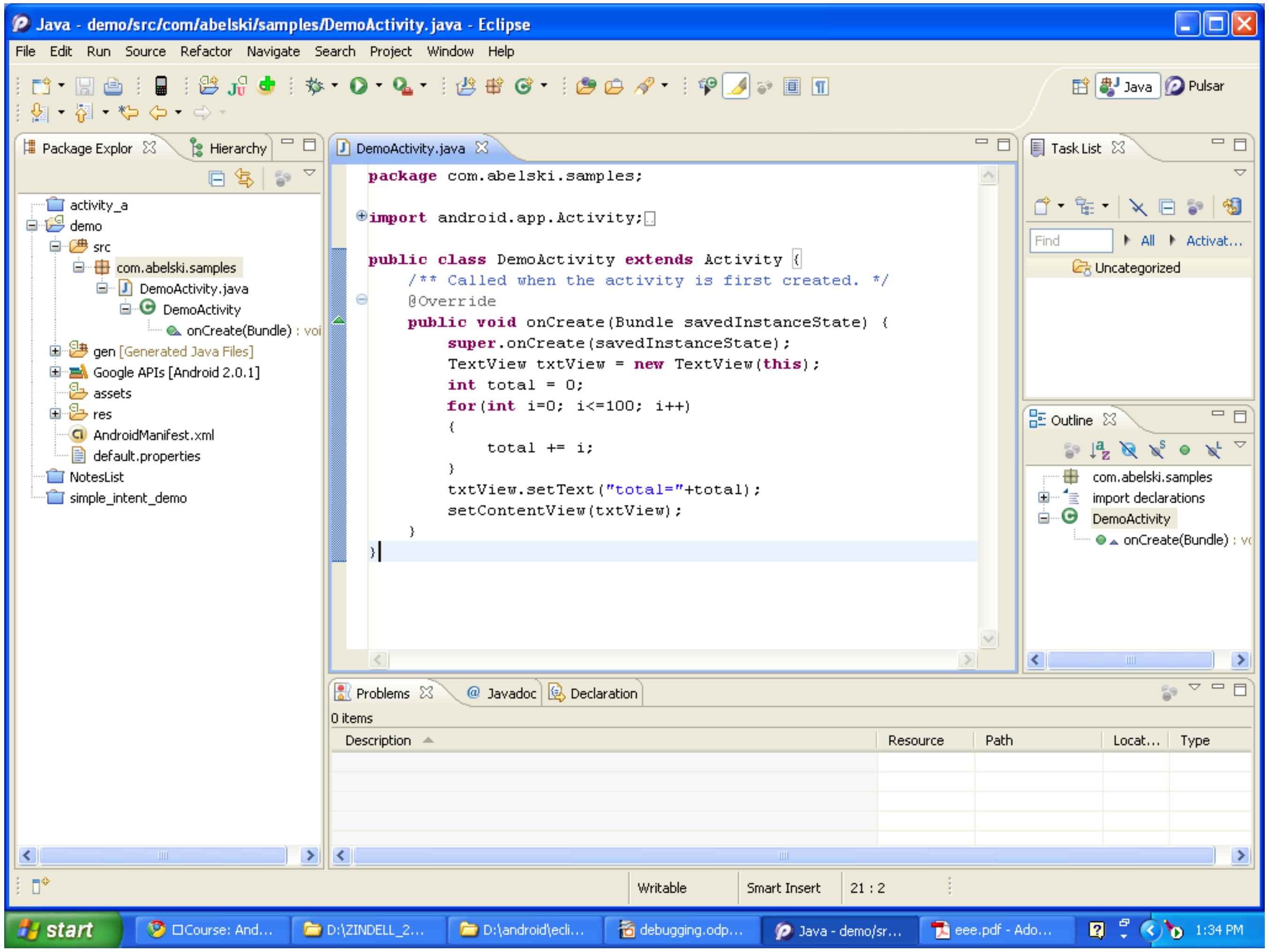
## Introduction

- The Eclipse IDE and the android SDK provide a rich set of tools that assist us with developing our application for the android platform and with debugging it.

## Eclipse Java Editor

- Developing for the android platform we can enjoy the same Eclipse IDE features we know when using the Eclipse IDE for other platforms.

# Eclipse Java Editor

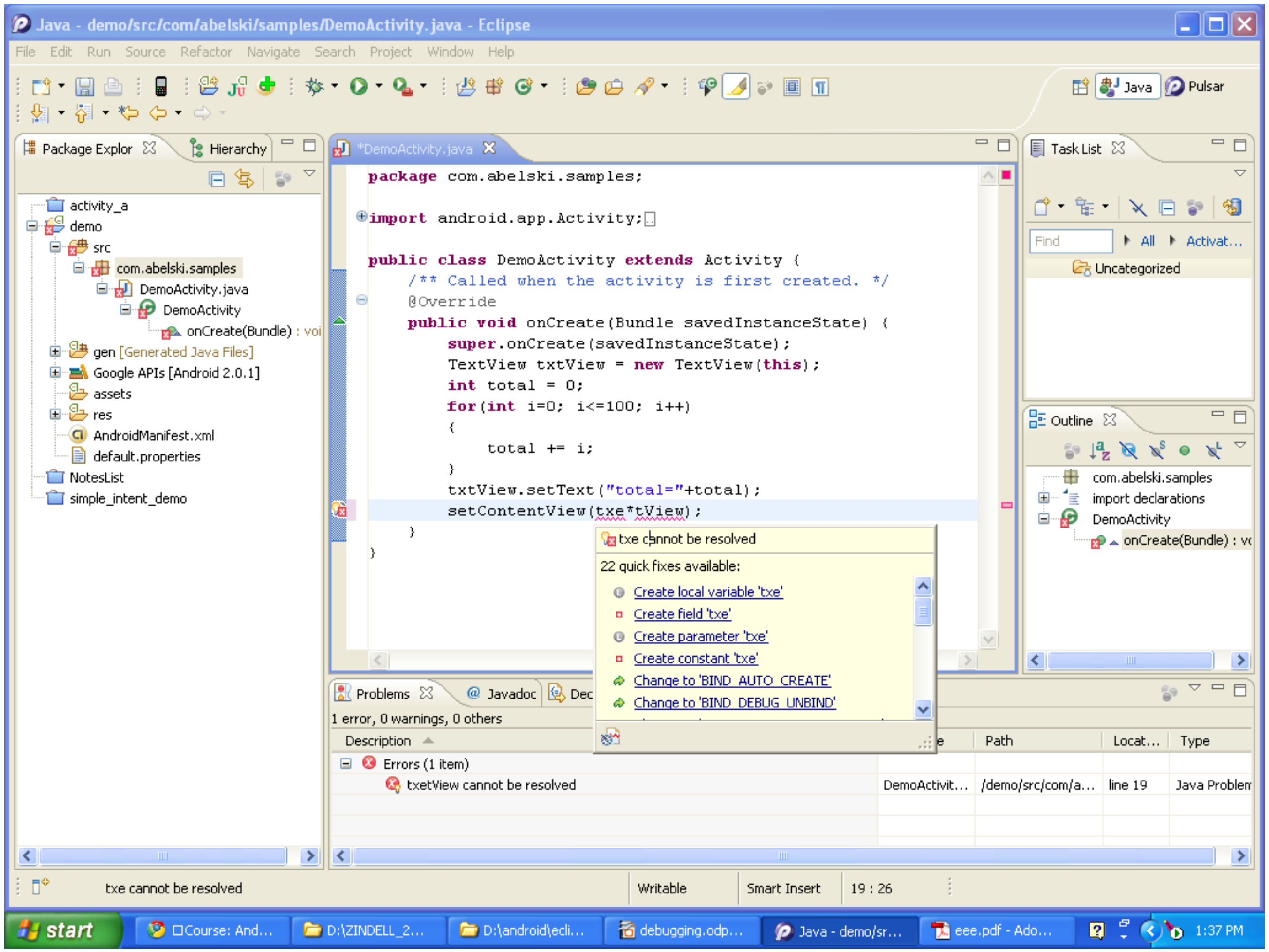


```
package com.abelski.samples;

import android.app.Activity;

public class DemoActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView txtView = new TextView(this);
        int total = 0;
        for(int i=0; i<=100; i++)
        {
            total += i;
        }
        txtView.setText("total="+total);
        setContentView(txtView);
    }
}
```

## Eclipse Java Editor

- When using the Eclipse IDE we get error messages on the fly concurrently with coding our program.
- We even get suggestions for fixing our code.

# Eclipse Java Editor



```
package com.abelski.samples;

import android.app.Activity;

public class DemoActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView txtView = new TextView(this);
        int total = 0;
        for(int i=0; i<=100; i++)
        {
            total += i;
        }
        txtView.setText("total="+total);
        setContentView(txe*tView);
    }
}
```

txe cannot be resolved

22 quick fixes available:

- Create local variable 'txe'
- Create field 'txe'
- Create parameter 'txe'
- Create constant 'txe'
- Change to 'BIND_AUTO_CREATE'
- Change to 'BIND_DEBUG_UNBIND'

1 error, 0 warnings, 0 others

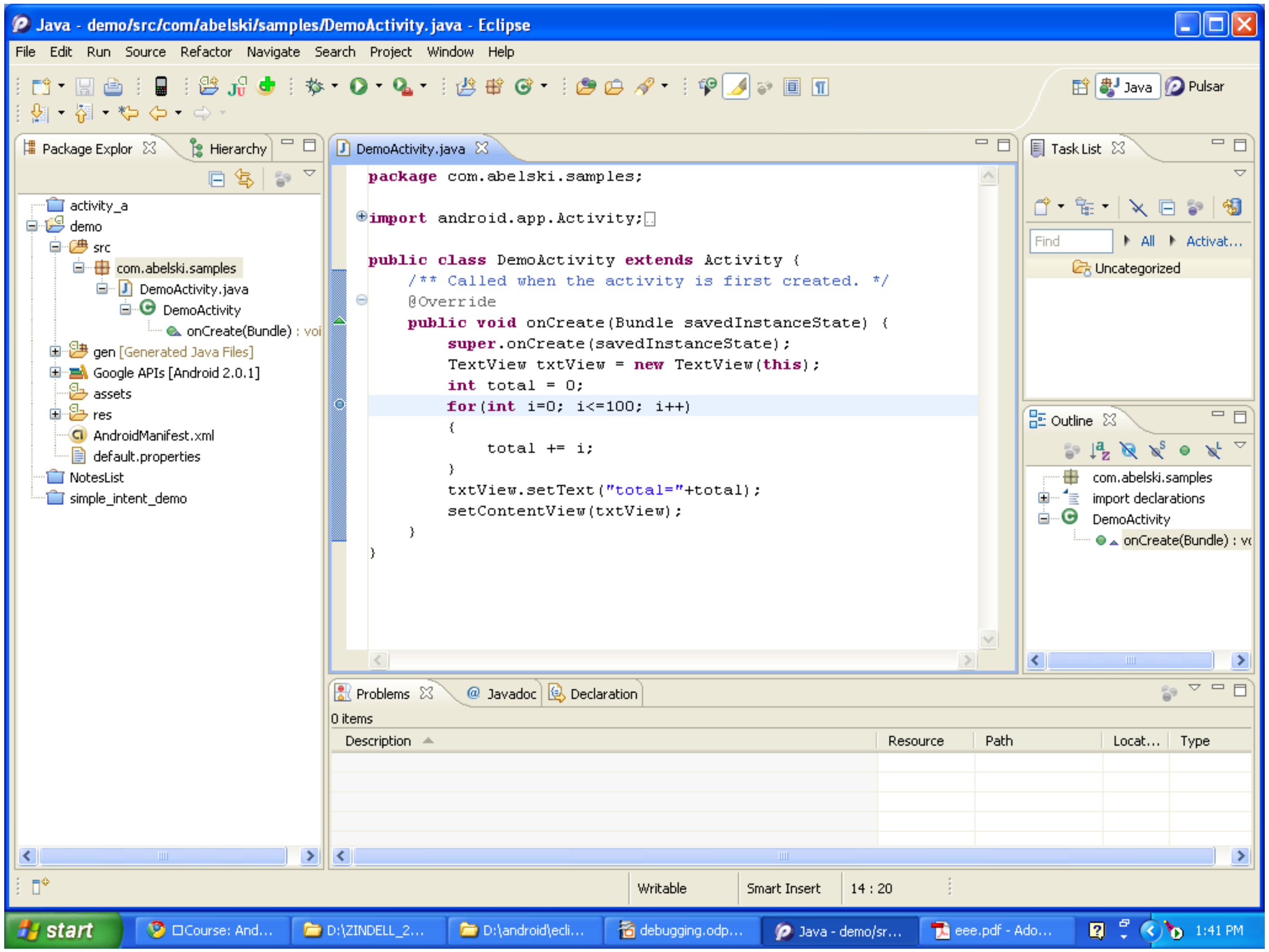
| Description | Resource | Path | Locat... | Type |
|---|---|---|---|---|
| Errors (1 item) | | | | |
| txetView cannot be resolved | DemoActivit... | /demo/src/com/a... | line 19 | Java Problem |

txe cannot be resolved | Writable | Smart Insert | 19 : 26

## Eclipse Java Editor

- There are three available ways for toggling a break-point.
- We can select the line and select from the top menu Run->Toggle Break Point
- We can double click in the left margin of the editor at the line we want to toggle.
- We can use the keyboard pressing Ctrl+shift+B.

# Eclipse Java Editor

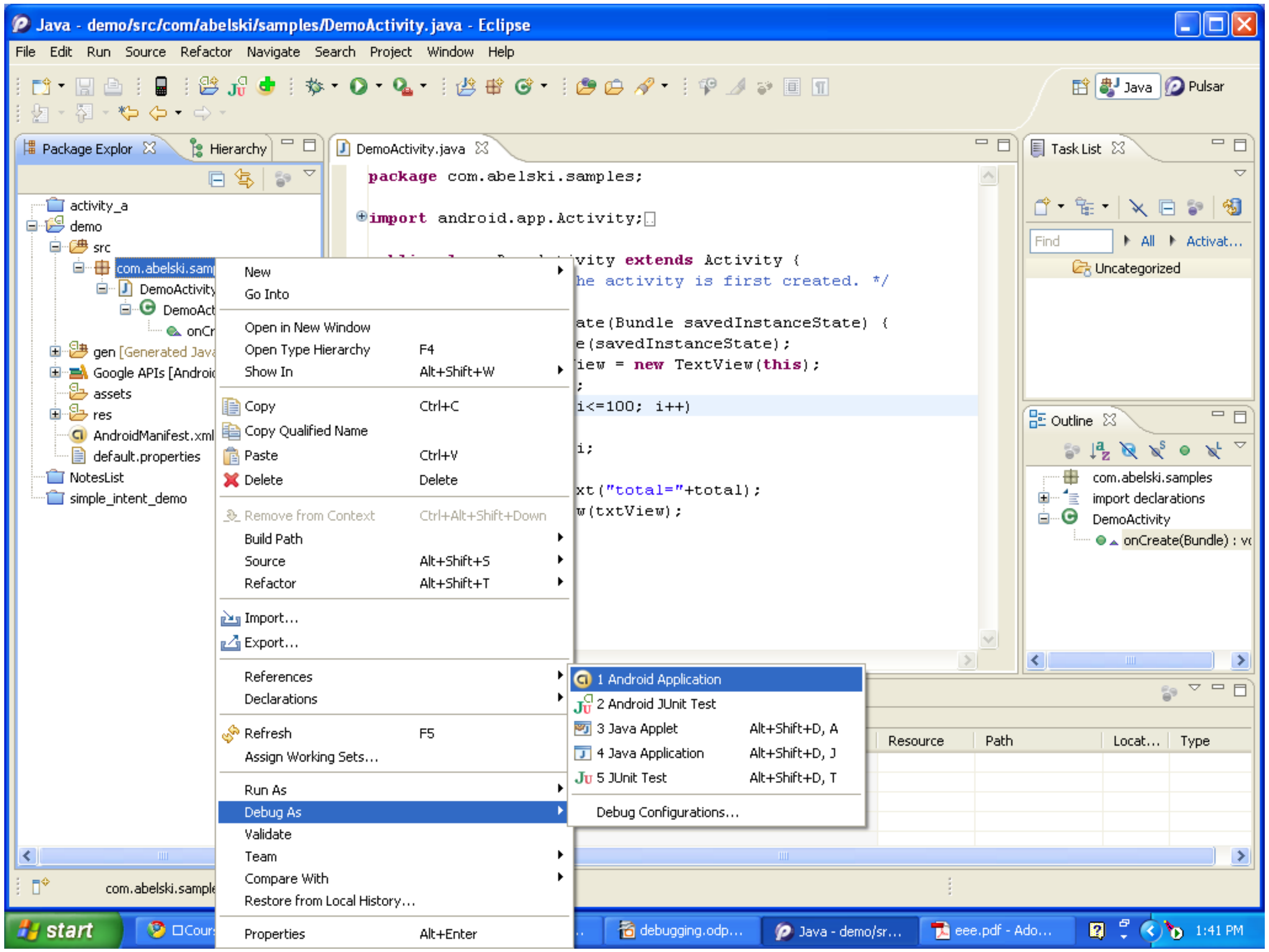


```
package com.abelski.samples;

import android.app.Activity;

public class DemoActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView txtView = new TextView(this);
        int total = 0;
        for(int i=0; i<=100; i++)
        {
            total += i;
        }
        txtView.setText("total="+total);
        setContentView(txtView);
    }
}
```

## Eclipse Java Debugger

- There are three available ways for toggling a break-point.
- We can select the line and select from the top menu `Run->Toggle Break Point`.
- We can double click in the left margin of the editor at the line we want to toggle.
- We can use the keyboard pressing `Ctrl+shift+B`.
- We start debugging by selecting from our top menu the '`Debug As`' option.

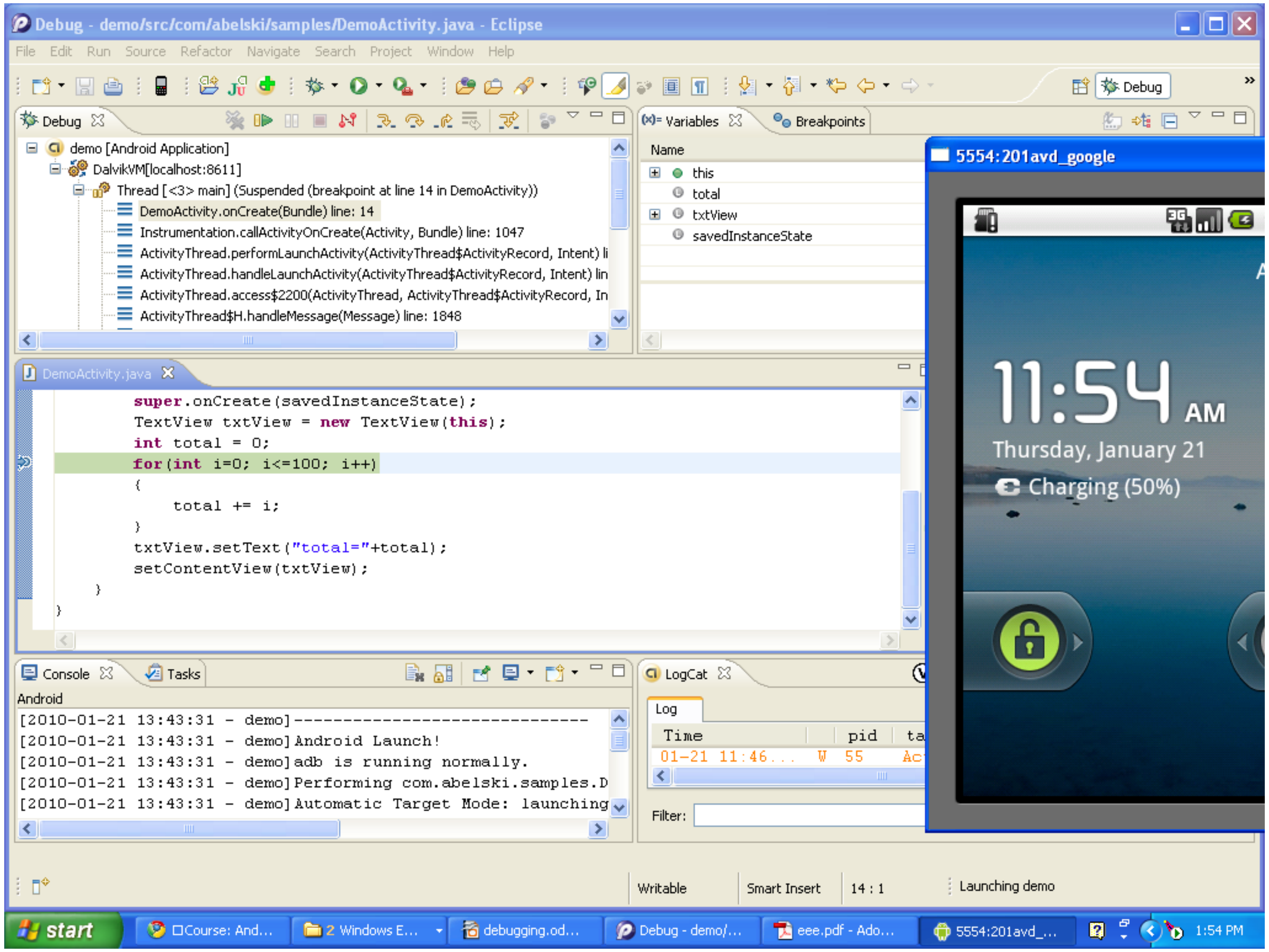
# Eclipse Java Debugger



```
package com.abelski.samples;

⊕import android.app.Activity;

                    vity extends Activity {
                    he activity is first created. */

                    ate(Bundle savedInstanceState) {
                    e(savedInstanceState);
                    iew = new TextView(this);
                    ;
                    i<=100; i++)

                    i;

                    xt("total="+total);
                    w(txtView);
```

© 2008 Haim Michael

# Eclipse Java Debugger



```
demo [Android Application]
  DalvikVM[localhost:8611]
    Thread [<3> main] (Suspended (breakpoint at line 14 in DemoActivity))
      DemoActivity.onCreate(Bundle) line: 14
      Instrumentation.callActivityOnCreate(Activity, Bundle) line: 1047
      ActivityThread.performLaunchActivity(ActivityThread$ActivityRecord, Intent) li
      ActivityThread.handleLaunchActivity(ActivityThread$ActivityRecord, Intent) lin
      ActivityThread.access$2200(ActivityThread, ActivityThread$ActivityRecord, In
      ActivityThread$H.handleMessage(Message) line: 1848
```

```
Name
this
total
txtView
savedInstanceState
```

```java
        super.onCreate(savedInstanceState);
        TextView txtView = new TextView(this);
        int total = 0;
        for(int i=0; i<=100; i++)
        {
            total += i;
        }
        txtView.setText("total="+total);
        setContentView(txtView);
    }
}
```

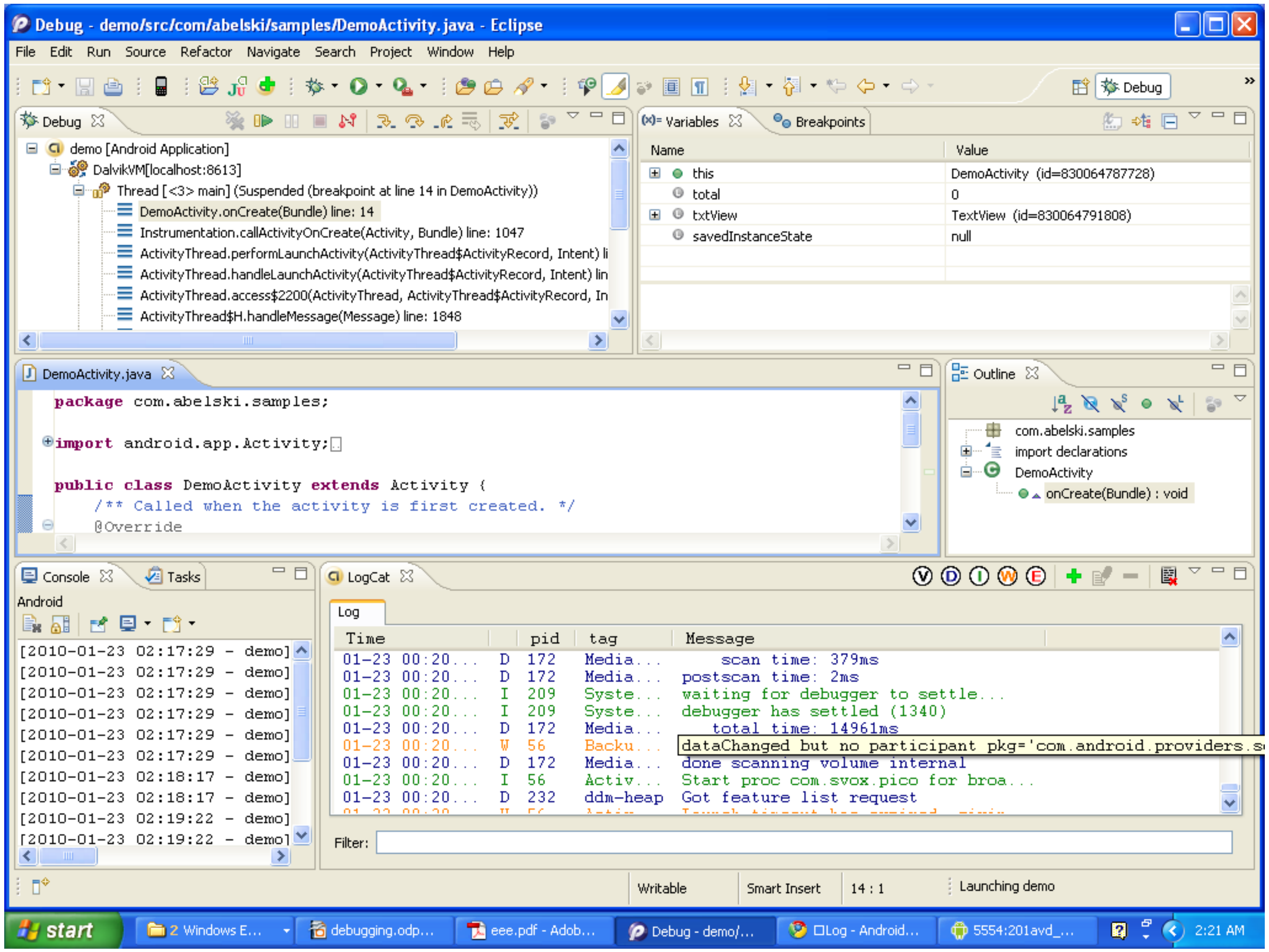
```
Android
[2010-01-21 13:43:31 - demo]------------------------------
[2010-01-21 13:43:31 - demo]Android Launch!
[2010-01-21 13:43:31 - demo]adb is running normally.
[2010-01-21 13:43:31 - demo]Performing com.abelski.samples.D
[2010-01-21 13:43:31 - demo]Automatic Target Mode: launching
```

```
Log
Time            pid  ta
01-21 11:46...  W  55  Ac
Filter:
```

## Logcat

- Logcat is a general purpose logging facility. The Logcat pane is available as part of the debugger perspective. The Logcat pane includes a log of messages.

# Logcat



```
demo [Android Application]
  DalvikVM[localhost:8613]
    Thread [<3> main] (Suspended (breakpoint at line 14 in DemoActivity))
      DemoActivity.onCreate(Bundle) line: 14
      Instrumentation.callActivityOnCreate(Activity, Bundle) line: 1047
      ActivityThread.performLaunchActivity(ActivityThread$ActivityRecord, Intent) li
      ActivityThread.handleLaunchActivity(ActivityThread$ActivityRecord, Intent) lin
      ActivityThread.access$2200(ActivityThread, ActivityThread$ActivityRecord, In
      ActivityThread$H.handleMessage(Message) line: 1848
```

| Name | Value |
|---|---|
| this | DemoActivity (id=830064787728) |
| total | 0 |
| txtView | TextView (id=830064791808) |
| savedInstanceState | null |

```java
package com.abelski.samples;

import android.app.Activity;

public class DemoActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
```

```
com.abelski.samples
  import declarations
  DemoActivity
    onCreate(Bundle) : void
```

```
Android
[2010-01-23 02:17:29 - demo]
[2010-01-23 02:17:29 - demo]
[2010-01-23 02:17:29 - demo]
[2010-01-23 02:17:29 - demo]
[2010-01-23 02:17:29 - demo]
[2010-01-23 02:17:29 - demo]
[2010-01-23 02:18:17 - demo]
[2010-01-23 02:18:17 - demo]
[2010-01-23 02:19:22 - demo]
[2010-01-23 02:19:22 - demo]
```

```
 Time             pid  tag        Message
01-23 00:20...  D  172  Media...      scan time: 379ms
01-23 00:20...  D  172  Media...  postscan time: 2ms
01-23 00:20...  I  209  Syste...  waiting for debugger to settle...
01-23 00:20...  I  209  Syste...  debugger has settled (1340)
01-23 00:20...  D  172  Media...     total time: 14961ms
01-23 00:20...  W  56   Backu...  dataChanged but no participant pkg='com.android.providers.s
01-23 00:20...  D  172  Media...  done scanning volume internal
01-23 00:20...  I  56   Activ...  Start proc com.svox.pico for broa...
01-23 00:20...  D  232  ddm-heap  Got feature list request
```

## Logcat

- Each one of the messages has a different entry priority. The Log class includes separated static methods for each one of the available entry priorities.

```
static int  d(String tag, String msg, Throwable tr)
```

Send a DEBUG log message and log the exception.

```
static int  d(String tag, String msg)
```

Send a DEBUG log message.

# Logcat

```
static int  e(String tag, String msg)
```

Send an ERROR log message.

```
static int  e(String tag, String msg, Throwable tr)
```

Send a ERROR log message and log the exception.

```
static int  i(String tag, String msg, Throwable tr)
```

Send a INFO log message and log the exception.

```
static int  i(String tag, String msg)
```

Send an INFO log message.

# Logcat

```
static int  v(String tag, String msg, Throwable tr)
```

Send a VERBOSE log message and log the exception.

```
static int  v(String tag, String msg)
```

Send a VERBOSE log message.

```
static int  w(String tag, String msg)
```

Send a WARN log message.

```
static int  w(String tag, Throwable tr)
```

Send a WARN log message.

## Logcat

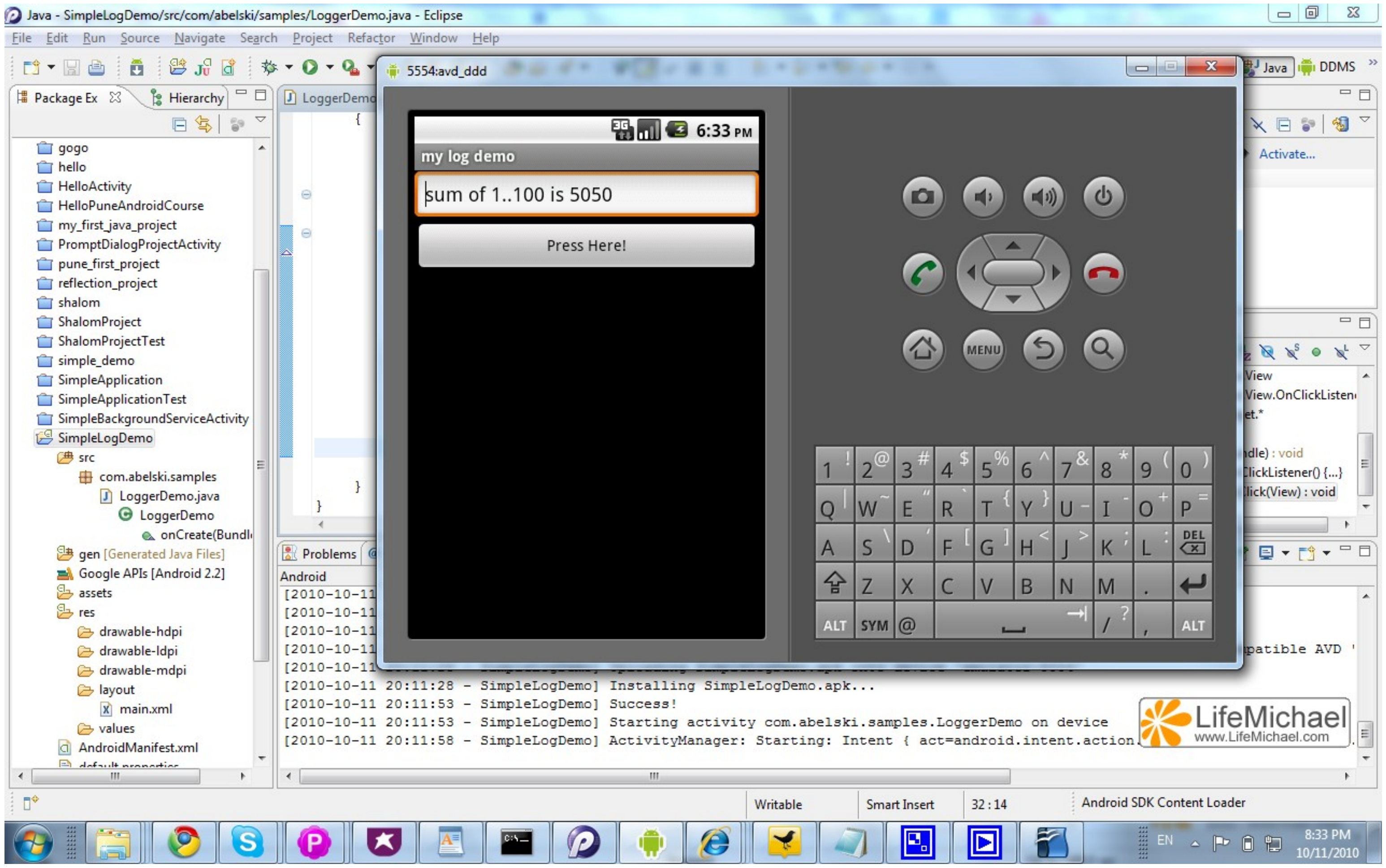
```
static int  w(String tag, String msg, Throwable tr)
```

Send a WARN log message and log the exception.

## Sample

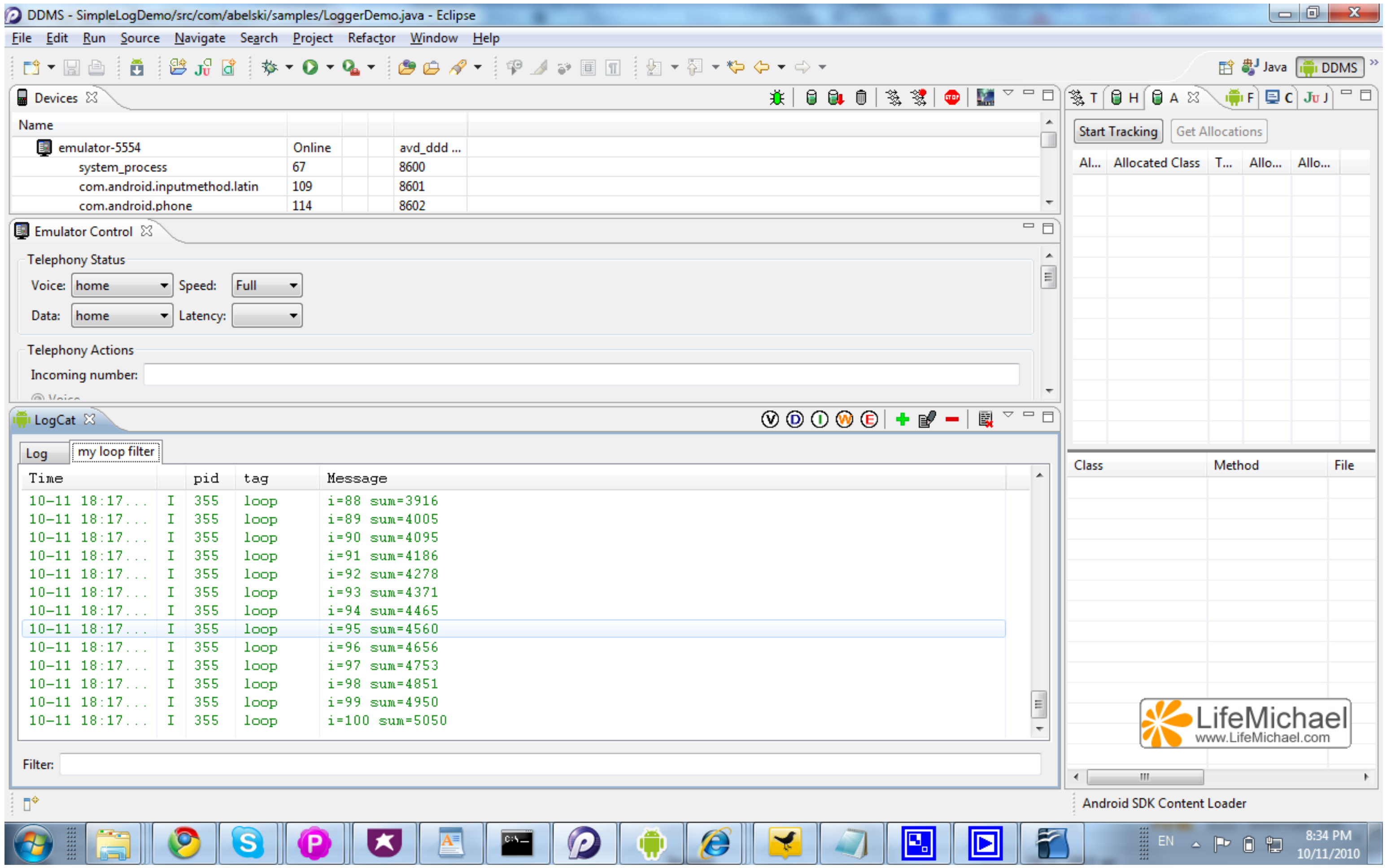
```
public class LoggerDemo extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button bt = (Button)findViewById(R.id.Button01);
        bt.setOnClickListener(new OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                int sum = 0;
                for(int i=1; i<=100; i++)
                {
                    sum += i;
                    Log.i("loop","i="+i+" sum="+sum);
                }
                EditText text = (EditText)findViewById(R.id.EditText01);
                text.setText("sum of 1..100 is "+sum);
            }
        });
    }
}
```

# Sample



```
[2010-10-11 20:11:28 - SimpleLogDemo] Installing SimpleLogDemo.apk...
[2010-10-11 20:11:53 - SimpleLogDemo] Success!
[2010-10-11 20:11:53 - SimpleLogDemo] Starting activity com.abelski.samples.LoggerDemo on device
[2010-10-11 20:11:58 - SimpleLogDemo] ActivityManager: Starting: Intent { act=android.intent.action
```

# Sample


DDMS - SimpleLogDemo/src/com/abelski/samples/LoggerDemo.java - Eclipse
File Edit Run Source Navigate Search Project Refactor Window Help
Java DDMS
Devices
Emulator Control
Telephony Status
Voice: home
Speed: Full
Data: home
Latency:
Telephony Actions
Incoming number:
LogCat
Log
my loop filter
Filter:
Start Tracking
Get Allocations
Al... Allocated Class T... Allo... Allo...
Class Method File
LifeMichael www.LifeMichael.com
Android SDK Content Loader

| Name | | |
|---|---|---|
| emulator-5554 | Online | avd_ddd ... |
| system_process | 67 | 8600 |
| com.android.inputmethod.latin | 109 | 8601 |
| com.android.phone | 114 | 8602 |

```
Time              pid  tag    Message
10-11 18:17...  I  355  loop   i=88 sum=3916
10-11 18:17...  I  355  loop   i=89 sum=4005
10-11 18:17...  I  355  loop   i=90 sum=4095
10-11 18:17...  I  355  loop   i=91 sum=4186
10-11 18:17...  I  355  loop   i=92 sum=4278
10-11 18:17...  I  355  loop   i=93 sum=4371
10-11 18:17...  I  355  loop   i=94 sum=4465
10-11 18:17...  I  355  loop   i=95 sum=4560
10-11 18:17...  I  355  loop   i=96 sum=4656
10-11 18:17...  I  355  loop   i=97 sum=4753
10-11 18:17...  I  355  loop   i=98 sum=4851
10-11 18:17...  I  355  loop   i=99 sum=4950
10-11 18:17...  I  355  loop   i=100 sum=5050
```

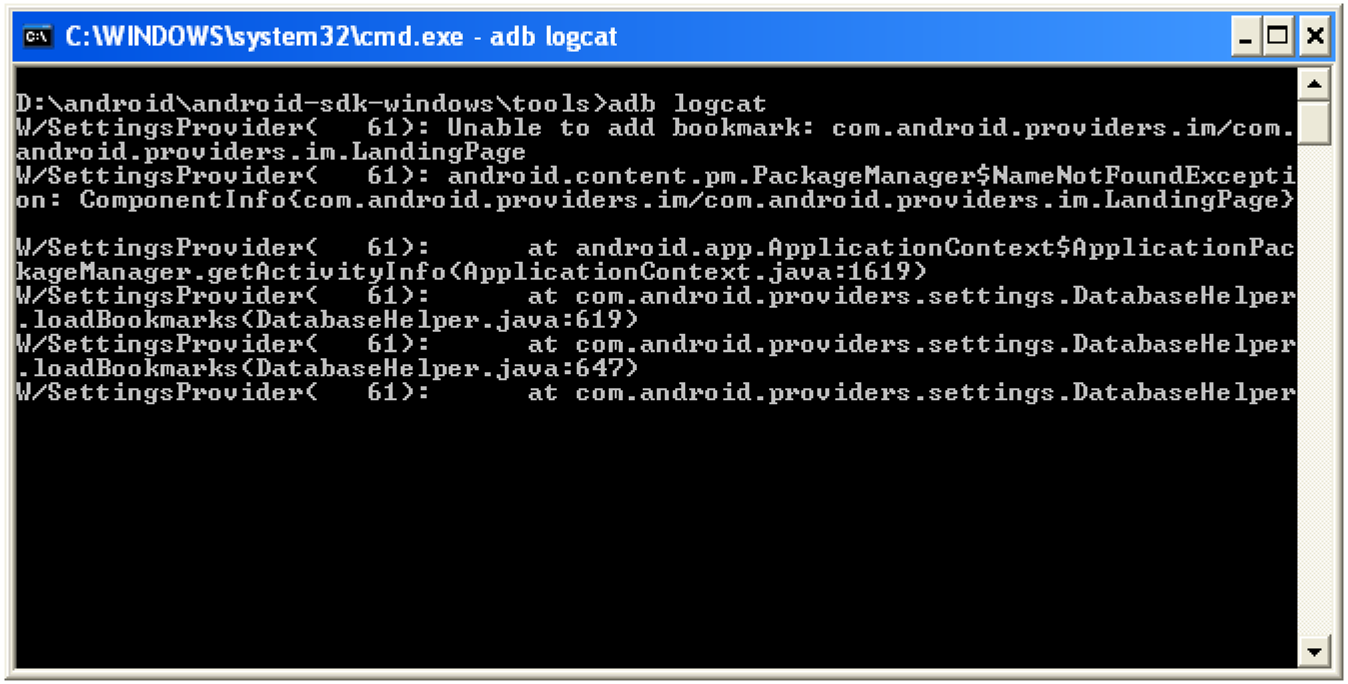
## Android Debug Bridge

- The android debug bridge (adb) is a special command line tool the android platform comes with.
- Using the android debug bridge we can remotely control the device or the emulator we are working with.
- We can invoke the android debug bridge client from the command line prompt.

## Android Debug Bridge

- Typing '`adb logcat`' in the command line will get us the detailed logcat messages.

## Android Debug Bridge

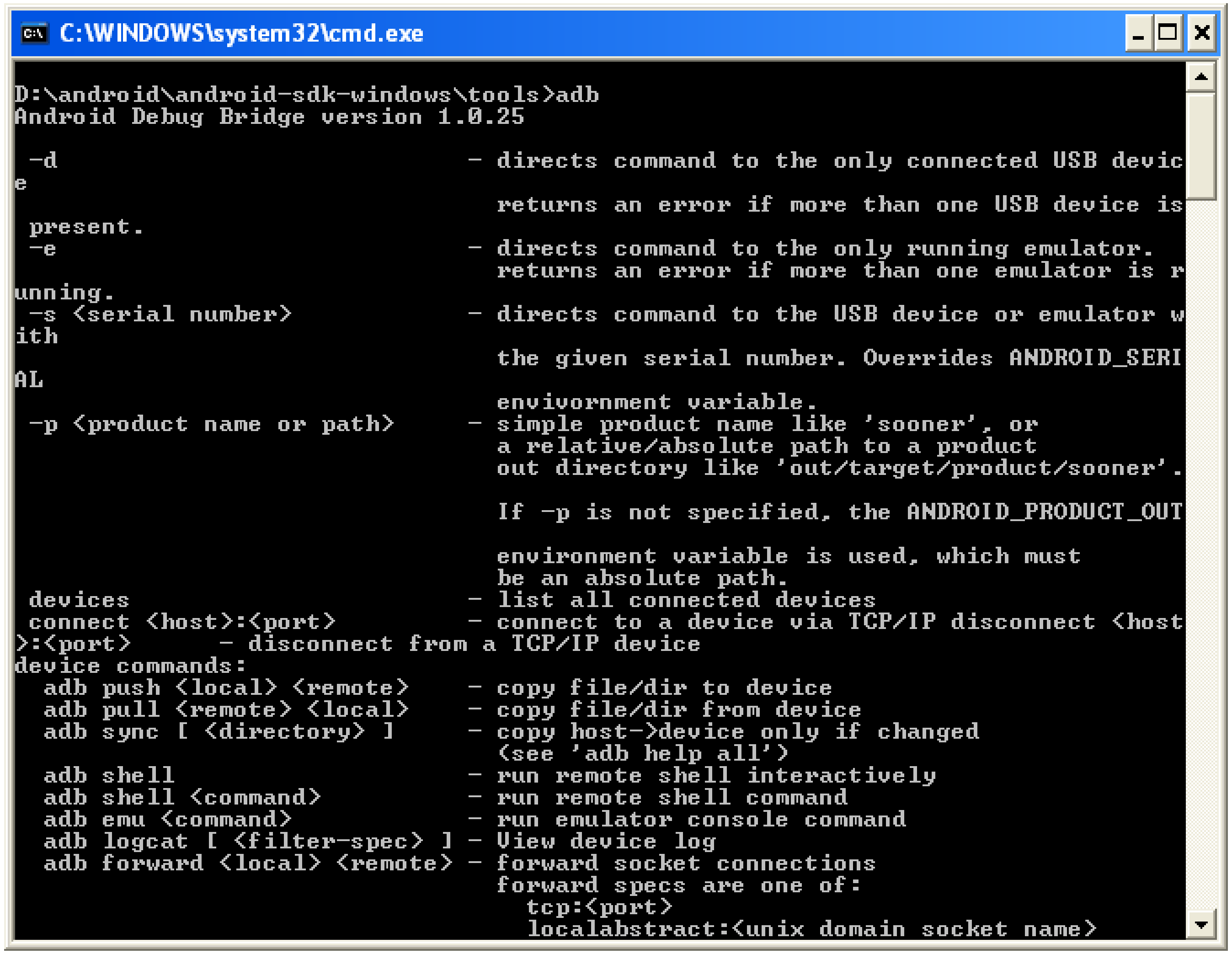

C:\WINDOWS\system32\cmd.exe - adb logcat

```
D:\android\android-sdk-windows\tools>adb logcat
W/SettingsProvider(   61): Unable to add bookmark: com.android.providers.im/com.
android.providers.im.LandingPage
W/SettingsProvider(   61): android.content.pm.PackageManager$NameNotFoundExcepti
on: ComponentInfo{com.android.providers.im/com.android.providers.im.LandingPage}

W/SettingsProvider(   61):      at android.app.ApplicationContext$ApplicationPac
kageManager.getActivityInfo(ApplicationContext.java:1619)
W/SettingsProvider(   61):      at com.android.providers.settings.DatabaseHelper
.loadBookmarks(DatabaseHelper.java:619)
W/SettingsProvider(   61):      at com.android.providers.settings.DatabaseHelper
.loadBookmarks(DatabaseHelper.java:647)
W/SettingsProvider(   61):      at com.android.providers.settings.DatabaseHelper
```

## Android Debug Bridge

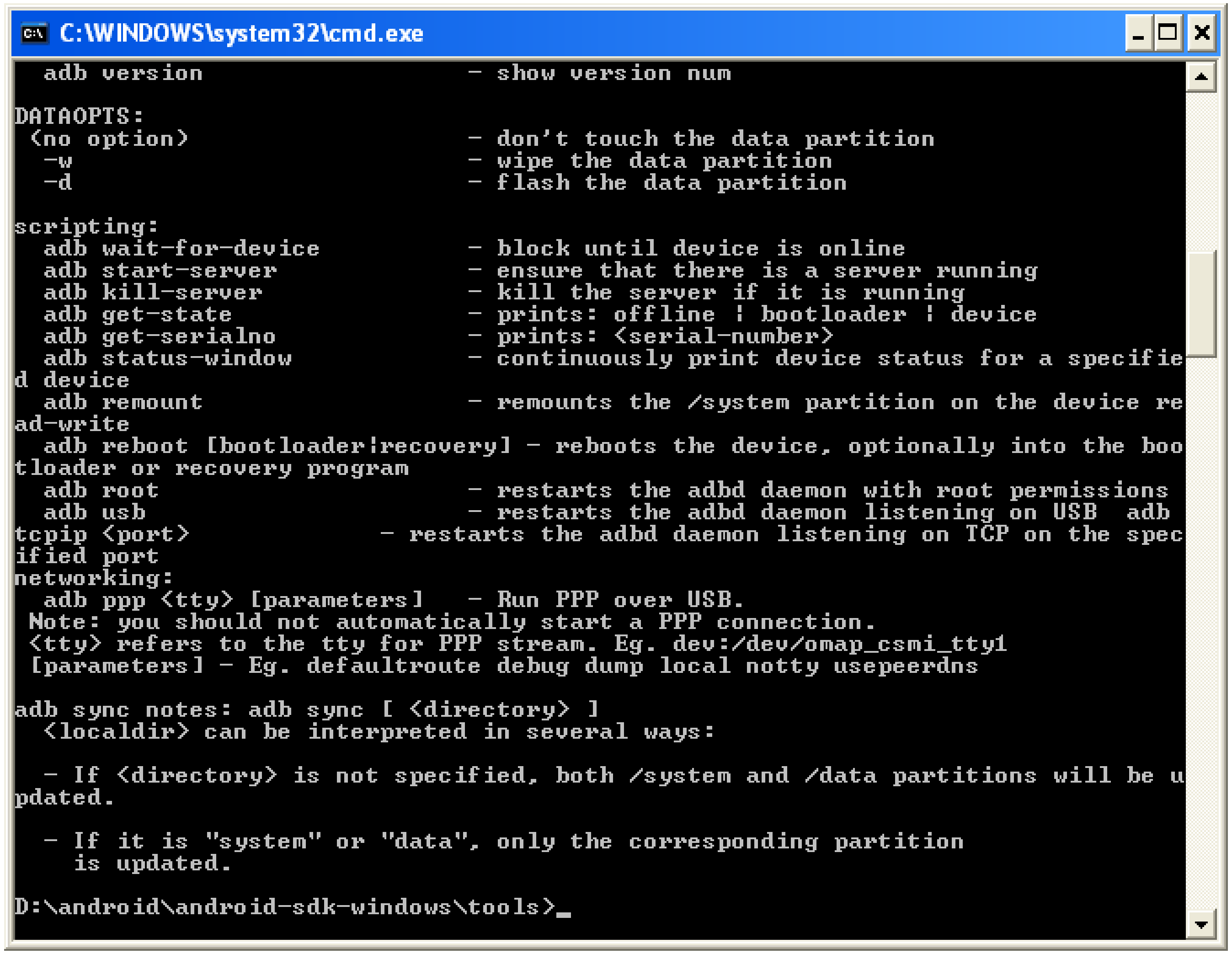
- Typing '`adb`' in the command line will get us a detailed list of all available `adb`'s commands.

# Android Debug Bridge



```
D:\android\android-sdk-windows\tools>adb
Android Debug Bridge version 1.0.25

 -d                            - directs command to the only connected USB devic
e
                                 returns an error if more than one USB device is
 present.
 -e                            - directs command to the only running emulator.
                                 returns an error if more than one emulator is r
unning.
 -s <serial number>            - directs command to the USB device or emulator w
ith
                                 the given serial number. Overrides ANDROID_SERI
AL
                                 envivornment variable.
 -p <product name or path>     - simple product name like 'sooner', or
                                 a relative/absolute path to a product
                                 out directory like 'out/target/product/sooner'.
                                 If -p is not specified, the ANDROID_PRODUCT_OUT
                                 environment variable is used, which must
                                 be an absolute path.
 devices                       - list all connected devices
 connect <host>:<port>         - connect to a device via TCP/IP disconnect <host
>:<port>        - disconnect from a TCP/IP device
device commands:
  adb push <local> <remote>    - copy file/dir to device
  adb pull <remote> <local>    - copy file/dir from device
  adb sync [ <directory> ]     - copy host->device only if changed
                                 (see 'adb help all')
  adb shell                    - run remote shell interactively
  adb shell <command>          - run remote shell command
  adb emu <command>            - run emulator console command
  adb logcat [ <filter-spec> ] - View device log
  adb forward <local> <remote> - forward socket connections
                                 forward specs are one of:
                                   tcp:<port>
                                   localabstract:<unix domain socket name>
```

# Android Debug Bridge



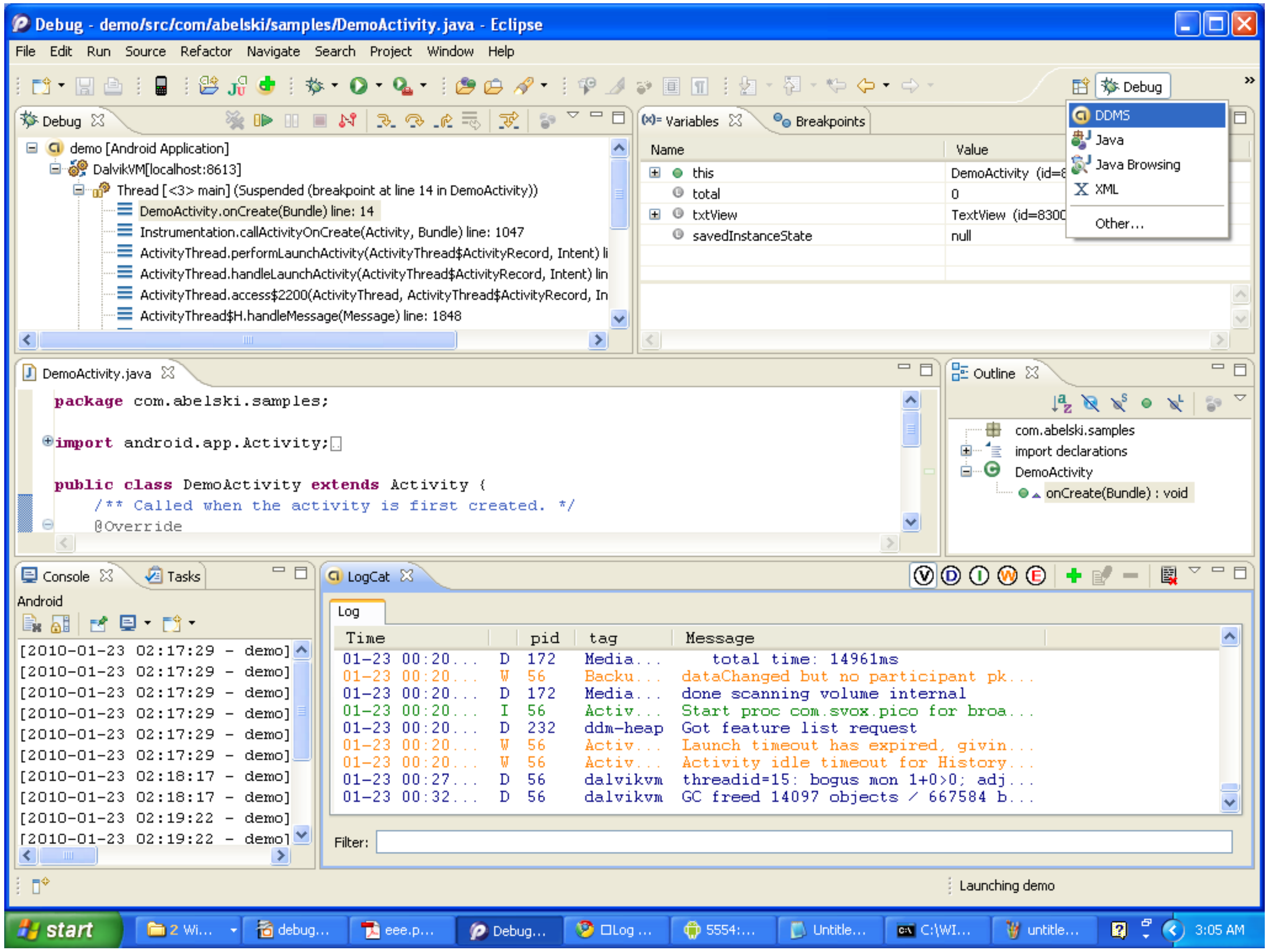
C:\WINDOWS\system32\cmd.exe

```
  adb version                    - show version num

DATAOPTS:
 (no option)                   - don't touch the data partition
  -w                           - wipe the data partition
  -d                           - flash the data partition

scripting:
  adb wait-for-device          - block until device is online
  adb start-server             - ensure that there is a server running
  adb kill-server              - kill the server if it is running
  adb get-state                - prints: offline | bootloader | device
  adb get-serialno             - prints: <serial-number>
  adb status-window            - continuously print device status for a specifie
d device
  adb remount                  - remounts the /system partition on the device re
ad-write
  adb reboot [bootloader|recovery] - reboots the device, optionally into the boo
tloader or recovery program
  adb root                     - restarts the adbd daemon with root permissions
  adb usb                      - restarts the adbd daemon listening on USB  adb
tcpip <port>             - restarts the adbd daemon listening on TCP on the spec
ified port
networking:
  adb ppp <tty> [parameters]   - Run PPP over USB.
 Note: you should not automatically start a PPP connection.
 <tty> refers to the tty for PPP stream. Eg. dev:/dev/omap_csmi_tty1
 [parameters] - Eg. defaultroute debug dump local notty usepeerdns

adb sync notes: adb sync [ <directory> ]
  <localdir> can be interpreted in several ways:

  - If <directory> is not specified, both /system and /data partitions will be u
pdated.

  - If it is "system" or "data", only the corresponding partition
    is updated.

D:\android\android-sdk-windows\tools>_
```

## Delvik Debug Monitor Service

- Once the android SDK plug-in for the Eclipse IDE is installed, we can start using the Delvik Debug Monitor Service (DDMS).
- The DDMS perspective provides a window based interface for android specific debug information on the emulator (or the real handset).

# Delvik Debug Monitor Service



```
demo [Android Application]
  DalvikVM[localhost:8613]
    Thread [<3> main] (Suspended (breakpoint at line 14 in DemoActivity))
      DemoActivity.onCreate(Bundle) line: 14
      Instrumentation.callActivityOnCreate(Activity, Bundle) line: 1047
      ActivityThread.performLaunchActivity(ActivityThread$ActivityRecord, Intent) li
      ActivityThread.handleLaunchActivity(ActivityThread$ActivityRecord, Intent) lin
      ActivityThread.access$2200(ActivityThread, ActivityThread$ActivityRecord, In
      ActivityThread$H.handleMessage(Message) line: 1848
```

| Name | Value |
|---|---|
| this | DemoActivity (id=8 |
| total | 0 |
| txtView | TextView (id=8300 |
| savedInstanceState | null |

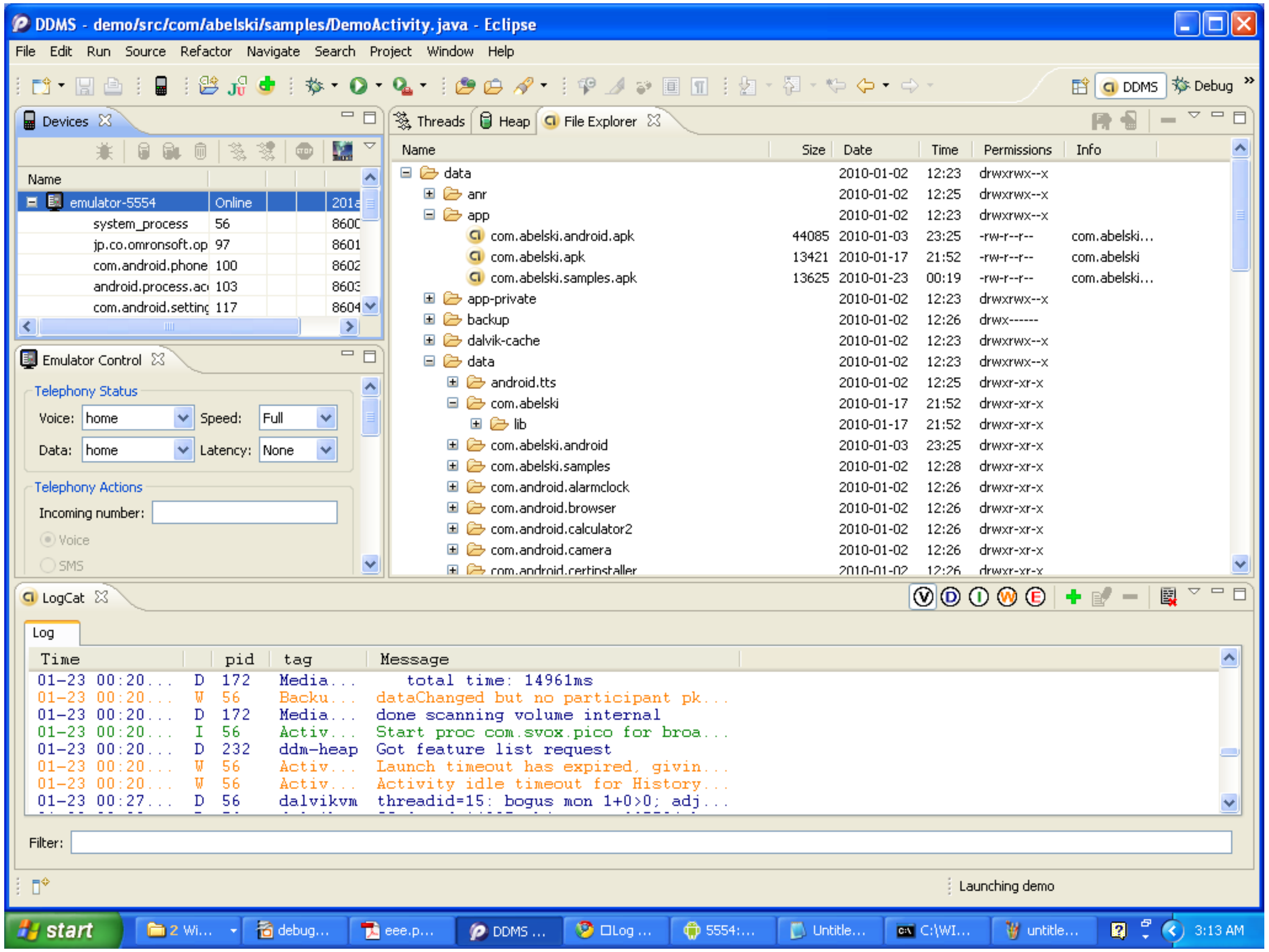
```
package com.abelski.samples;

import android.app.Activity;

public class DemoActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
```

```
com.abelski.samples
import declarations
DemoActivity
  onCreate(Bundle) : void
```

```
[2010-01-23 02:17:29 - demo]
[2010-01-23 02:17:29 - demo]
[2010-01-23 02:17:29 - demo]
[2010-01-23 02:17:29 - demo]
[2010-01-23 02:17:29 - demo]
[2010-01-23 02:17:29 - demo]
[2010-01-23 02:18:17 - demo]
[2010-01-23 02:18:17 - demo]
[2010-01-23 02:19:22 - demo]
[2010-01-23 02:19:22 - demo]
```

```
Time              pid  tag        Message
01-23 00:20...  D  172  Media...     total time: 14961ms
01-23 00:20...  W  56   Backu...   dataChanged but no participant pk...
01-23 00:20...  D  172  Media...   done scanning volume internal
01-23 00:20...  I  56   Activ...   Start proc com.svox.pico for broa...
01-23 00:20...  D  232  ddm-heap   Got feature list request
01-23 00:20...  W  56   Activ...   Launch timeout has expired, givin...
01-23 00:20...  W  56   Activ...   Activity idle timeout for History...
01-23 00:27...  D  56   dalvikvm   threadid=15: bogus mon 1+0>0; adj...
01-23 00:32...  D  56   dalvikvm   GC freed 14097 objects / 667584 b...
```

# Delvik Debug Monitor Service


DDMS - demo/src/com/abelski/samples/DemoActivity.java - Eclipse
File Edit Run Source Refactor Navigate Search Project Window Help
DDMS
Debug
Devices
Threads
Heap
File Explorer
Emulator Control
LogCat
Log
start
Launching demo
3:13 AM

| Name | | |
|---|---|---|
| emulator-5554 | Online | 201a |
| system_process | 56 | 8600 |
| jp.co.omronsoft.op | 97 | 8601 |
| com.android.phone | 100 | 8602 |
| android.process.ac | 103 | 8603 |
| com.android.setting | 117 | 8604 |

Telephony Status

Voice: home Speed: Full

Data: home Latency: None

Telephony Actions

Incoming number:

Voice

SMS

| Name | Size | Date | Time | Permissions | Info |
|---|---|---|---|---|---|
| data | | 2010-01-02 | 12:23 | drwxrwx--x | |
| anr | | 2010-01-02 | 12:25 | drwxrwx--x | |
| app | | 2010-01-02 | 12:23 | drwxrwx--x | |
| com.abelski.android.apk | 44085 | 2010-01-03 | 23:25 | -rw-r--r-- | com.abelski... |
| com.abelski.apk | 13421 | 2010-01-17 | 21:52 | -rw-r--r-- | com.abelski |
| com.abelski.samples.apk | 13625 | 2010-01-23 | 00:19 | -rw-r--r-- | com.abelski... |
| app-private | | 2010-01-02 | 12:23 | drwxrwx--x | |
| backup | | 2010-01-02 | 12:26 | drwx------ | |
| dalvik-cache | | 2010-01-02 | 12:23 | drwxrwx--x | |
| data | | 2010-01-02 | 12:23 | drwxrwx--x | |
| android.tts | | 2010-01-02 | 12:25 | drwxr-xr-x | |
| com.abelski | | 2010-01-17 | 21:52 | drwxr-xr-x | |
| lib | | 2010-01-17 | 21:52 | drwxr-xr-x | |
| com.abelski.android | | 2010-01-03 | 23:25 | drwxr-xr-x | |
| com.abelski.samples | | 2010-01-02 | 12:28 | drwxr-xr-x | |
| com.android.alarmclock | | 2010-01-02 | 12:26 | drwxr-xr-x | |
| com.android.browser | | 2010-01-02 | 12:26 | drwxr-xr-x | |
| com.android.calculator2 | | 2010-01-02 | 12:26 | drwxr-xr-x | |
| com.android.camera | | 2010-01-02 | 12:26 | drwxr-xr-x | |
| com.android.certinstaller | | 2010-01-02 | 12:26 | drwxr-xr-x | |

```
 Time            pid  tag        Message
01-23 00:20...  D  172  Media...    total time: 14961ms
01-23 00:20...  W  56   Backu...   dataChanged but no participant pk...
01-23 00:20...  D  172  Media...   done scanning volume internal
01-23 00:20...  I  56   Activ...   Start proc com.svox.pico for broa...
01-23 00:20...  D  232  ddm-heap   Got feature list request
01-23 00:20...  W  56   Activ...   Launch timeout has expired, givin...
01-23 00:20...  W  56   Activ...   Activity idle timeout for History...
01-23 00:27...  D  56   dalvikvm   threadid=15: bogus mon 1+0>0; adj...
```

Filter:

## Traceview

- The Traceview utility allows tracking the exact methods that are been called as well as the exact time each one of these methods executions takes.

## Traceview

- The Traceview utility includes two parts. The first is a small utility that creates a log file that includes detailed data about each and every method invocation. The second is a graphics based application you can execute passing over the log file and get a detailed graphics representation of all methods calls.

## Traceview

❖ In order to use the Traceview utility we should first import the `android.os` package and add to our code the following lines.

```
...
Debug.startMethodTracing(“mytrace”);
...

...
Debug.stopMethodTracing();
...
```

## Traceview

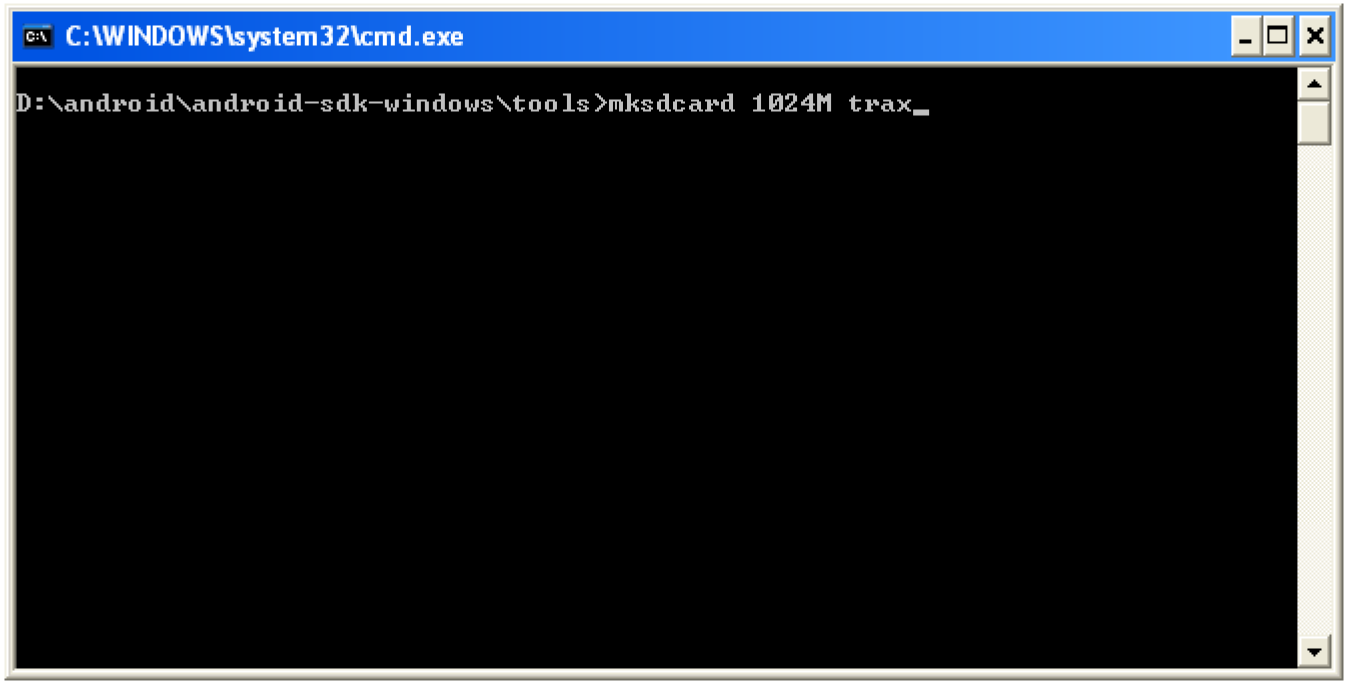
- The Tracebiew utility will fill in the log file on the android SD card with data generated during the code execution.
- If we work with the emulator then we should create an AVD with a virtual SD card.

## Traceview

- The first step should be creating the file on the SD card:

```
mksdcard 10M traco
```

## Traceview

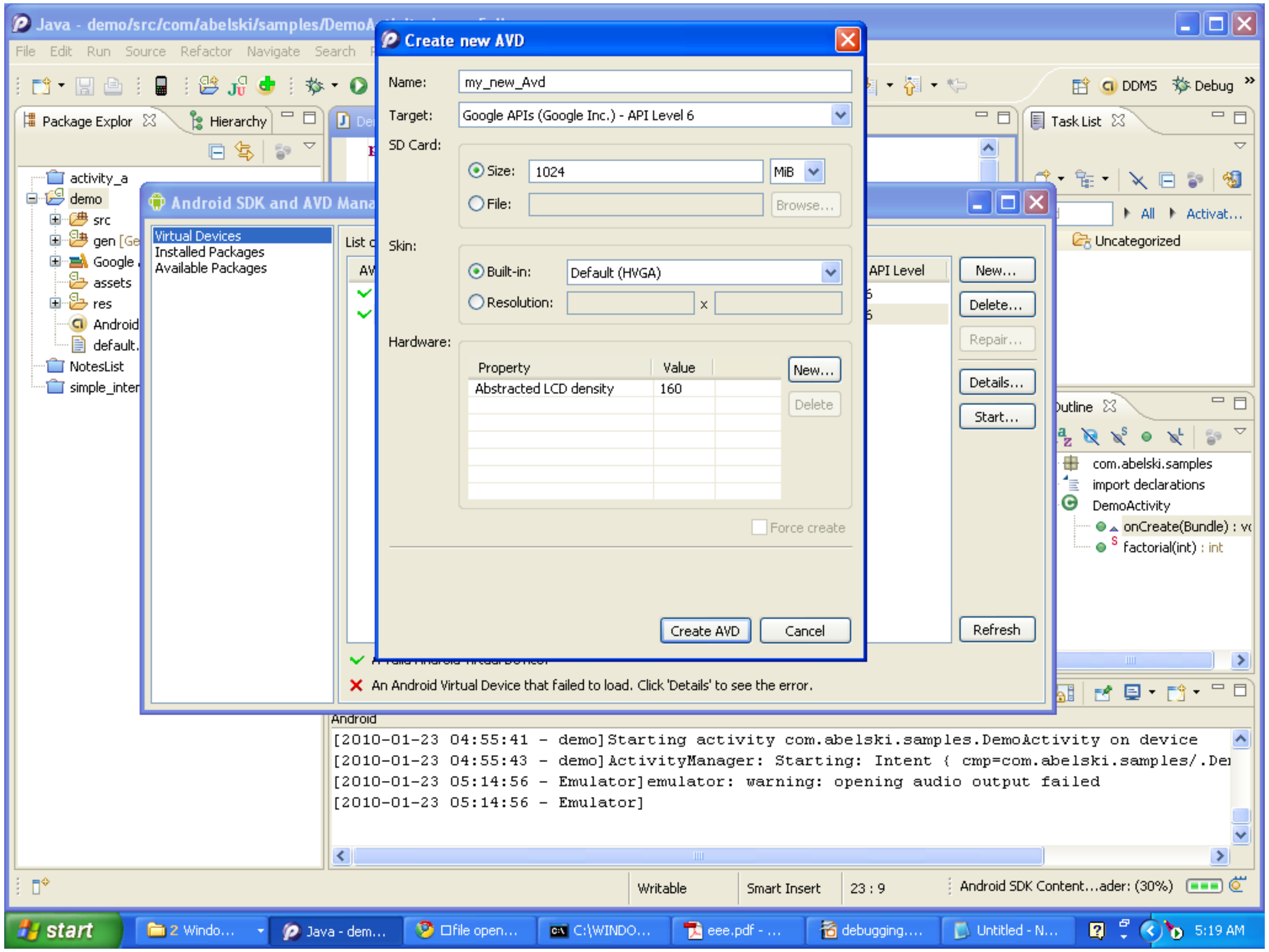


```
D:\android\android-sdk-windows\tools>mksdcard 1024M trax_
```

## Traceview

- The second step should be creating the AVD we want to use specifying the size of the requested SD card.

# Traceview



Create new AVD

Name: my_new_Avd

Target: Google APIs (Google Inc.) - API Level 6

SD Card: Size: 1024 MiB

Skin: Built-in: Default (HVGA)

Hardware:

| Property | Value |
|---|---|
| Abstracted LCD density | 160 |

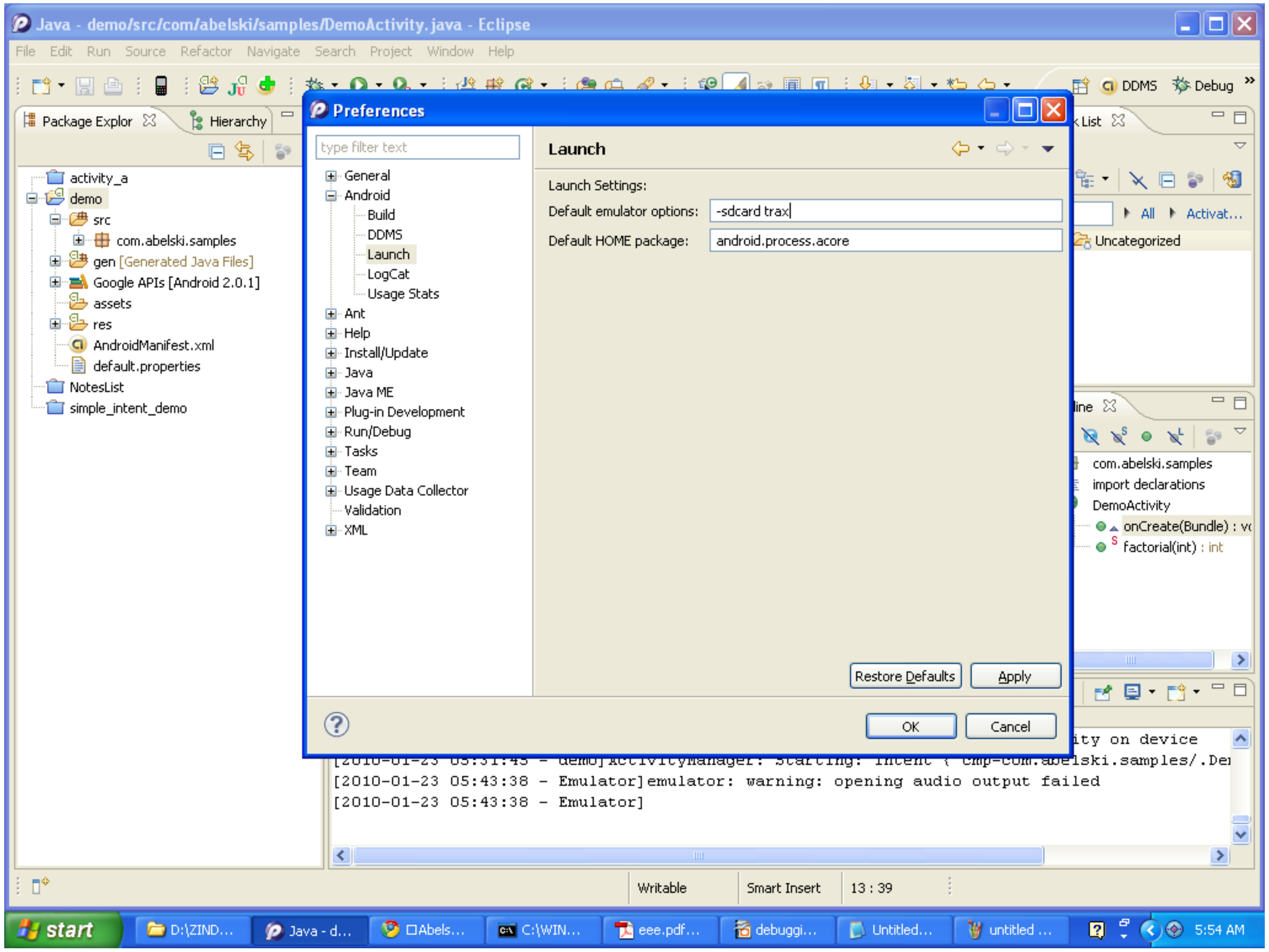
An Android Virtual Device that failed to load. Click 'Details' to see the error.

```
[2010-01-23 04:55:41 - demo]Starting activity com.abelski.samples.DemoActivity on device
[2010-01-23 04:55:43 - demo]ActivityManager: Starting: Intent { cmp=com.abelski.samples/.Der
[2010-01-23 05:14:56 - Emulator]emulator: warning: opening audio output failed
[2010-01-23 05:14:56 - Emulator]
```

## Traceview

- The third step should be telling the emulator we want to use a virtual SC card. In the Eclipse IDE you should choose `Window > Preferences > Android > Launch`. Within the box for the emulator options you should add the following code `-sdcard ./traco`
- Make sure to specify the complete path to the file, so the emulator can always find it.

# Traceview



Java - demo/src/com/abelski/samples/DemoActivity.java - Eclipse
Preferences
Launch

Launch Settings:

Default emulator options: -sdcard trax

Default HOME package: android.process.acore

```
[2010-01-23 05:31:45 - demo] ActivityManager: Starting: Intent { cmp=com.abelski.samples/.De
[2010-01-23 05:43:38 - Emulator] emulator: warning: opening audio output failed
[2010-01-23 05:43:38 - Emulator]
```

# Traceview

❖ The fourth step would be executing our code. Our code should include the call to start tracing the methods.

```
Debug.startMethodTracing("traco"); //traco is the filename
```

In addition, our code should include a call to stop it.

```
Debug.stopMethodTracing();
```

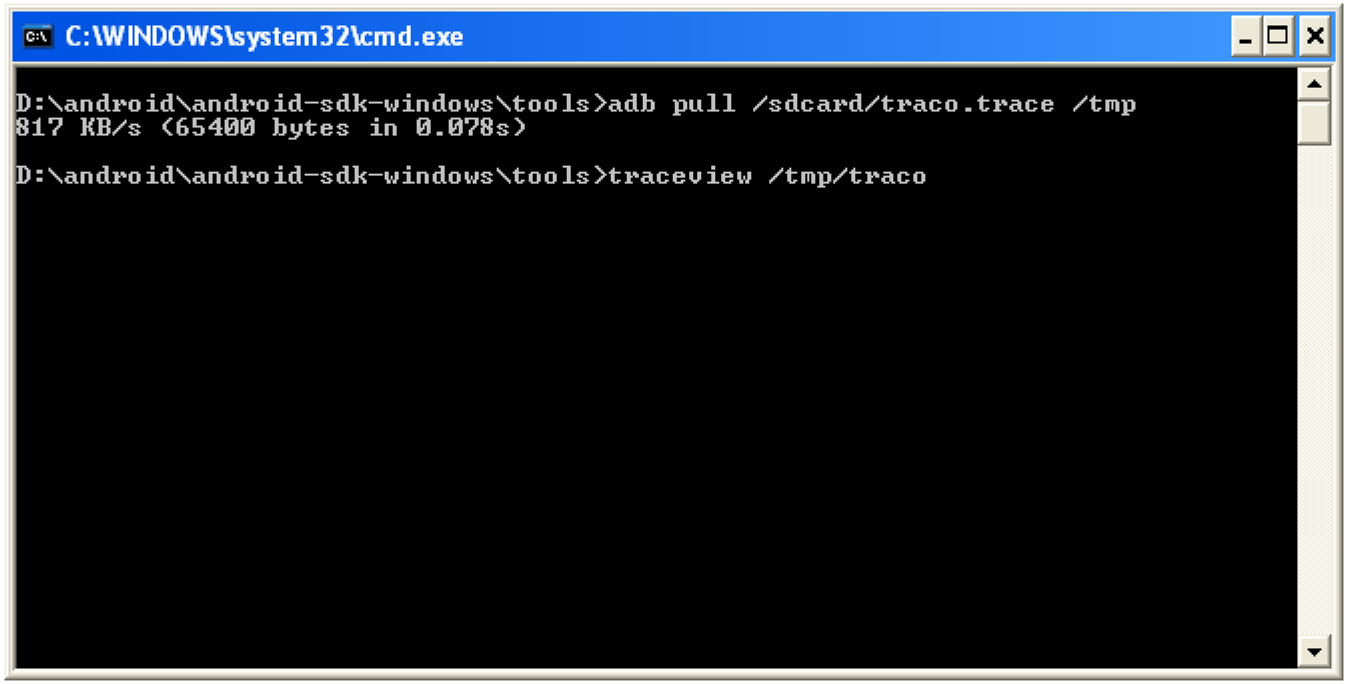
❖ When the application calls `startMethodTracing()`, the system creates a file called ___.trace (e.g. traco.trace). This file contains the binary method trace data and its mapping table with thread and method names.

## Traceview

- When using the Traceview utility the execution times are significantly slower. Therefore, we shouldn't refer these times as the accurate one. We can only compare them with each other.
- Once the execution completes we can get the log data displayed in a graphics way.

```
adb pull /sdcard/traco.trace /tmp
traceview /tmp/traco
```

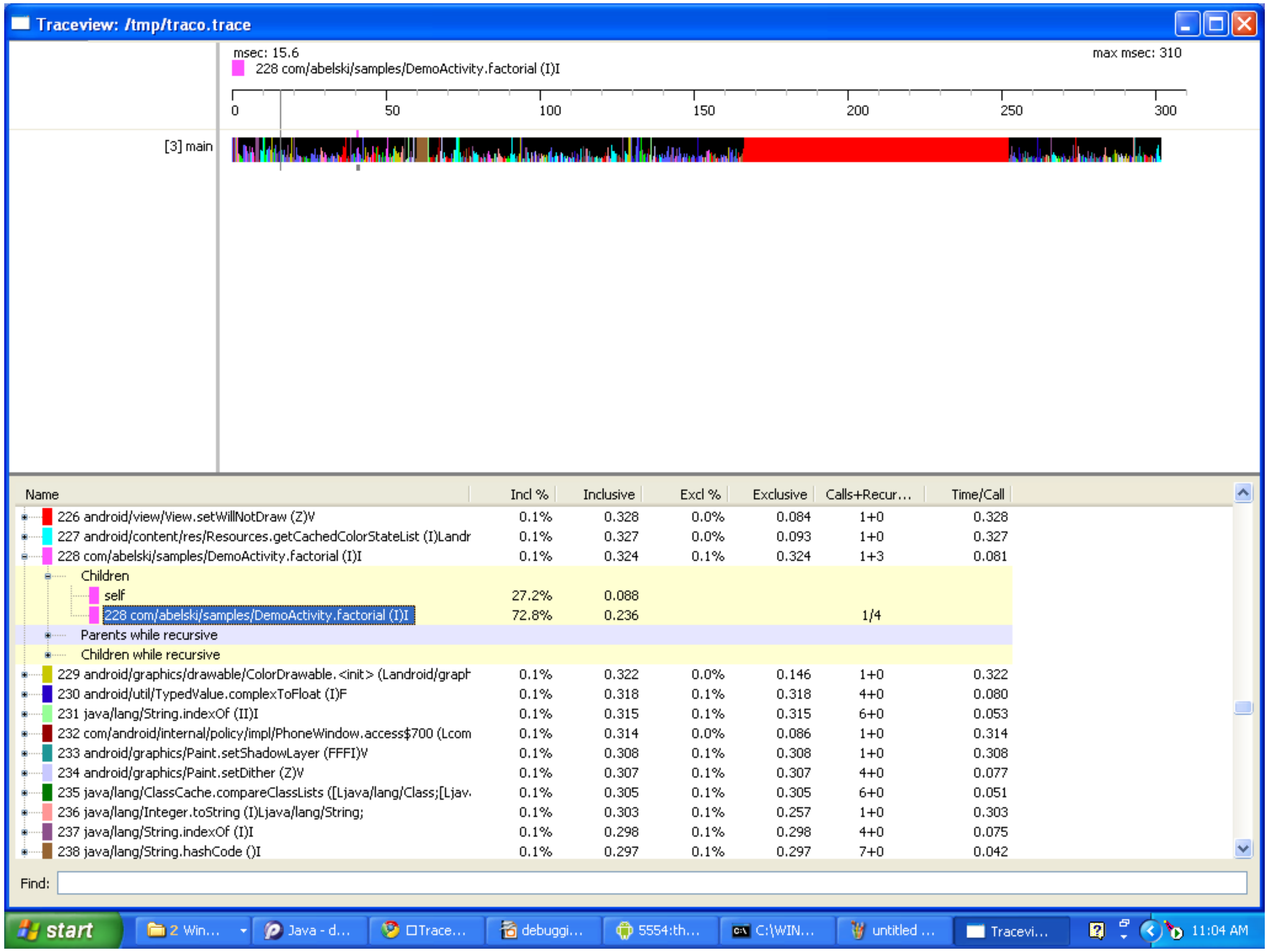
## Traceview



C:\WINDOWS\system32\cmd.exe

```
D:\android\android-sdk-windows\tools>adb pull /sdcard/traco.trace /tmp
817 KB/s (65400 bytes in 0.078s)

D:\android\android-sdk-windows\tools>traceview /tmp/traco
```

# Traceview



Traceview: /tmp/traco.trace

msec: 15.6 max msec: 310

228 com/abelski/samples/DemoActivity.factorial (I)I

[3] main

| Name | Incl % | Inclusive | Excl % | Exclusive | Calls+Recur... | Time/Call |
|---|---|---|---|---|---|---|
| 226 android/view/View.setWillNotDraw (Z)V | 0.1% | 0.328 | 0.0% | 0.084 | 1+0 | 0.328 |
| 227 android/content/res/Resources.getCachedColorStateList (I)Landr | 0.1% | 0.327 | 0.0% | 0.093 | 1+0 | 0.327 |
| 228 com/abelski/samples/DemoActivity.factorial (I)I | 0.1% | 0.324 | 0.1% | 0.324 | 1+3 | 0.081 |
| Children | | | | | | |
| self | 27.2% | 0.088 | | | | |
| 228 com/abelski/samples/DemoActivity.factorial (I)I | 72.8% | 0.236 | | | 1/4 | |
| Parents while recursive | | | | | | |
| Children while recursive | | | | | | |
| 229 android/graphics/drawable/ColorDrawable.<init> (Landroid/graph | 0.1% | 0.322 | 0.0% | 0.146 | 1+0 | 0.322 |
| 230 android/util/TypedValue.complexToFloat (I)F | 0.1% | 0.318 | 0.1% | 0.318 | 4+0 | 0.080 |
| 231 java/lang/String.indexOf (II)I | 0.1% | 0.315 | 0.1% | 0.315 | 6+0 | 0.053 |
| 232 com/android/internal/policy/impl/PhoneWindow.access$700 (Lcom | 0.1% | 0.314 | 0.0% | 0.086 | 1+0 | 0.314 |
| 233 android/graphics/Paint.setShadowLayer (FFFI)V | 0.1% | 0.308 | 0.1% | 0.308 | 1+0 | 0.308 |
| 234 android/graphics/Paint.setDither (Z)V | 0.1% | 0.307 | 0.1% | 0.307 | 4+0 | 0.077 |
| 235 java/lang/ClassCache.compareClassLists ([Ljava/lang/Class;[Ljav | 0.1% | 0.305 | 0.1% | 0.305 | 6+0 | 0.051 |
| 236 java/lang/Integer.toString (I)Ljava/lang/String; | 0.1% | 0.303 | 0.1% | 0.257 | 1+0 | 0.303 |
| 237 java/lang/String.indexOf (I)I | 0.1% | 0.298 | 0.1% | 0.298 | 4+0 | 0.075 |
| 238 java/lang/String.hashCode ()I | 0.1% | 0.297 | 0.1% | 0.297 | 7+0 | 0.042 |

## Debugging

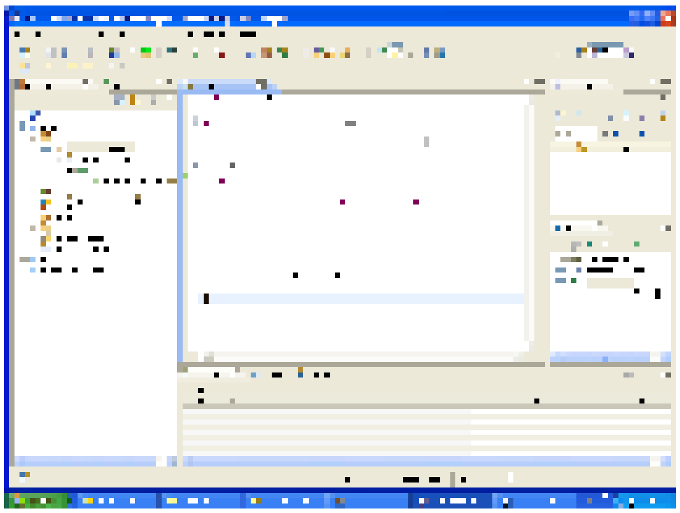
## Introduction

❖ The Eclipse IDE and the android SDK provide a rich set of tools that assist us with developing our application for the android platform and with debugging it.

## Eclipse Java Editor

❖ Developing for the android platform we can enjoy the same Eclipse IDE features we know when using the Eclipse IDE for other platforms.

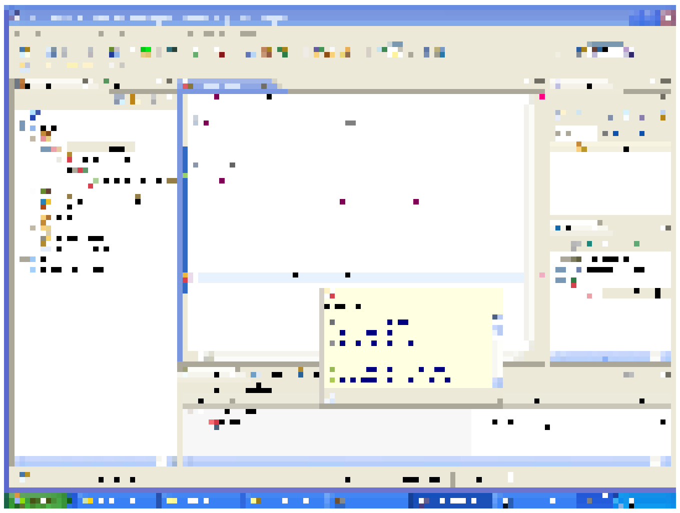
## Eclipse Java Editor



## Eclipse Java Editor

- When using the Eclipse IDE we get error messages on the fly concurrently with coding our program.
- We even get suggestions for fixing our code.

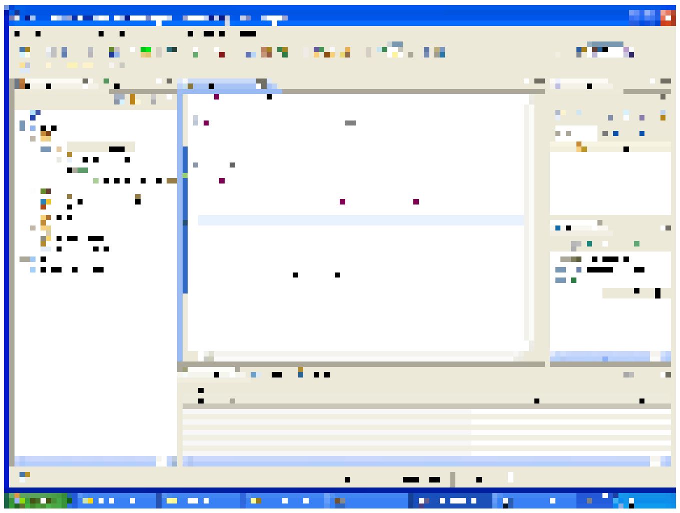
## Eclipse Java Editor

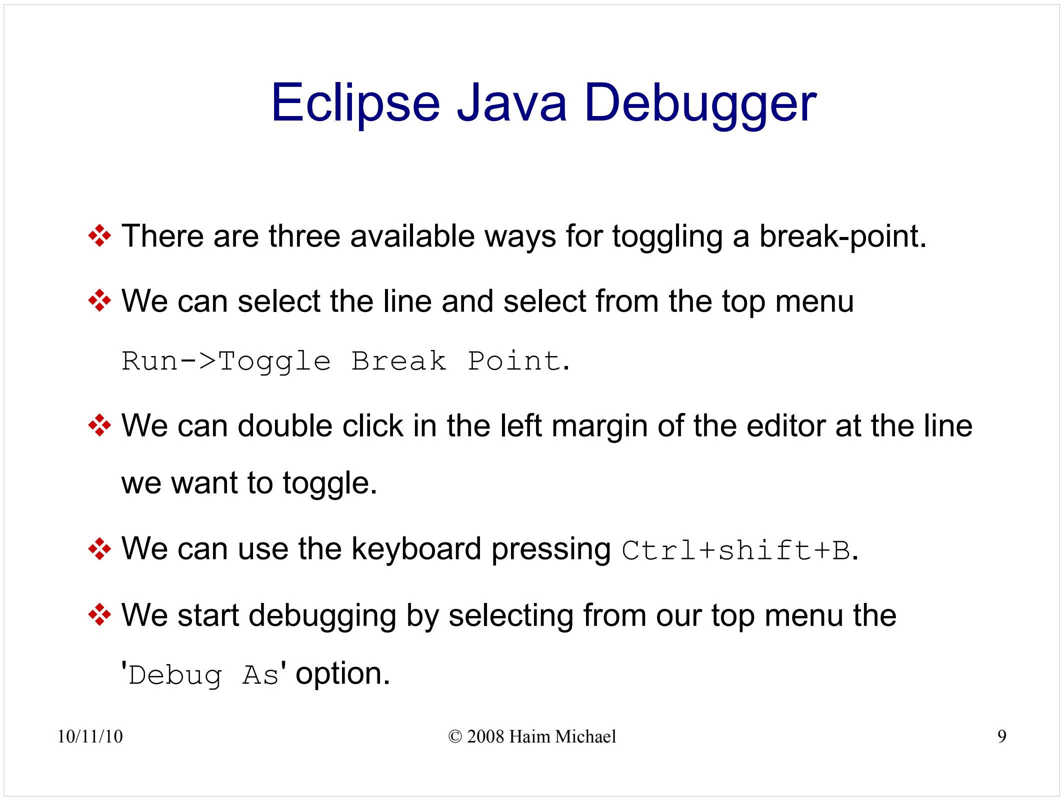


## Eclipse Java Editor

- There are three available ways for toggling a break-point.
- We can select the line and select from the top menu Run->Toggle Break Point
- We can double click in the left margin of the editor at the line we want to toggle.
- We can use the keyboard pressing Ctrl+shift+B.

## Eclipse Java Editor



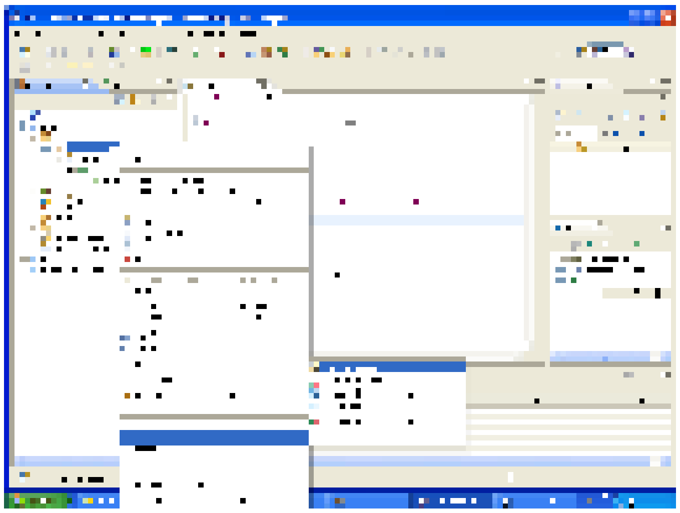


## Eclipse Java Debugger

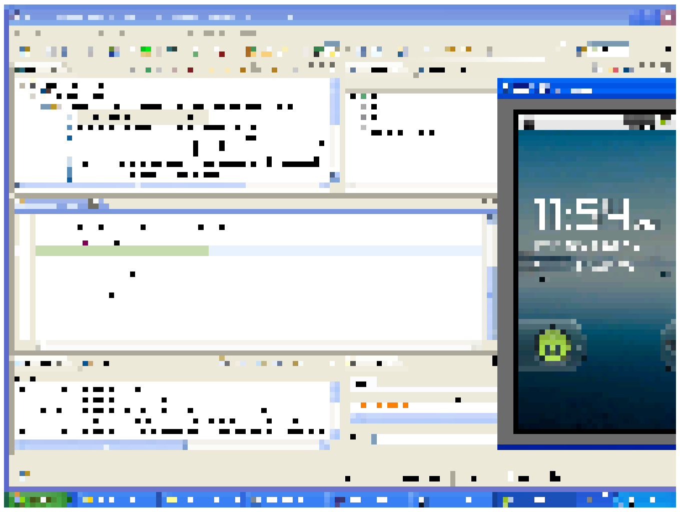
- There are three available ways for toggling a break-point.
- We can select the line and select from the top menu `Run->Toggle Break Point`.
- We can double click in the left margin of the editor at the line we want to toggle.
- We can use the keyboard pressing `Ctrl+shift+B`.
- We start debugging by selecting from our top menu the '`Debug As`' option.

10/11/10 © 2008 Haim Michael 9

## Eclipse Java Debugger



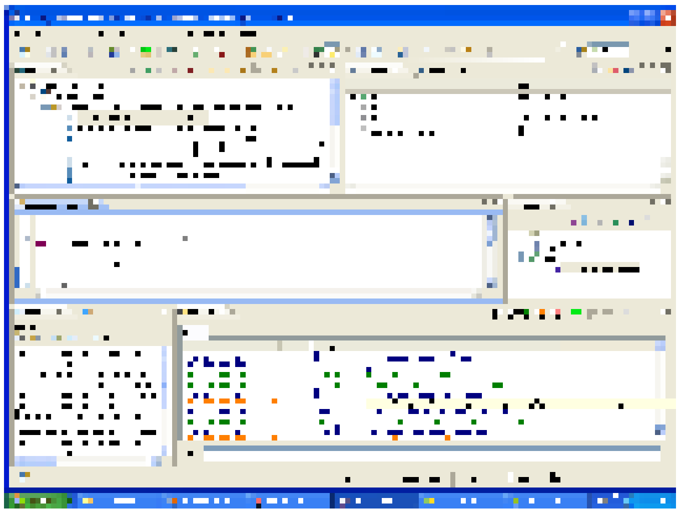
## Eclipse Java Debugger



## Logcat

❖ Logcat is a general purpose logging facility. The Logcat pane is available as part of the debugger perspective. The Logcat pane includes a log of messages.

## Logcat



## Logcat

- Each one of the messages has a different entry priority. The Log class includes separated static methods for each one of the available entry priorities.

```
static int  d(String tag, String msg, Throwable tr)
```

Send a DEBUG log message and log the exception.

```
static int  d(String tag, String msg)
```

Send a DEBUG log message.

## Logcat

```
static int  e(String tag, String msg)
```

Send an ERROR log message.

```
static int  e(String tag, String msg, Throwable tr)
```

Send a ERROR log message and log the exception.

```
static int  i(String tag, String msg, Throwable tr)
```

Send a INFO log message and log the exception.

```
static int  i(String tag, String msg)
```

Send an INFO log message.

## Logcat

```
static int  v(String tag, String msg, Throwable tr)
```

Send a VERBOSE log message and log the exception.

```
static int  v(String tag, String msg)
```

Send a VERBOSE log message.

```
static int  w(String tag, String msg)
```

Send a WARN log message.

```
static int  w(String tag, Throwable tr)
```

Send a WARN log message.

## Logcat

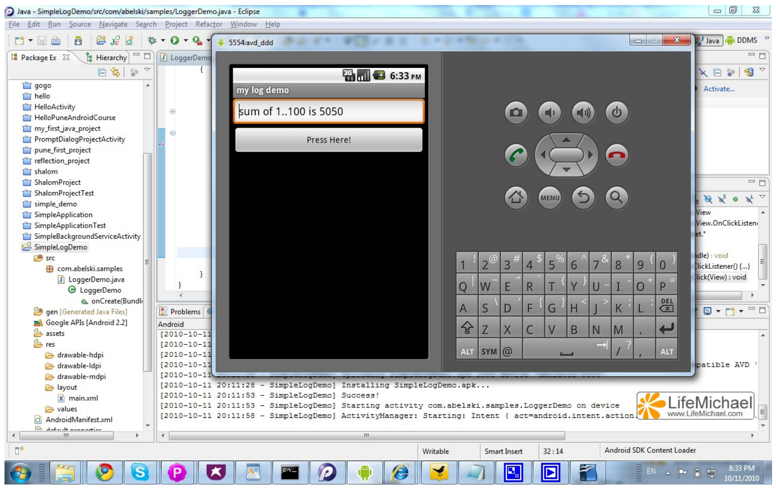
```
static int  w(String tag, String msg, Throwable tr)
```

Send a WARN log message and log the exception.

## Sample

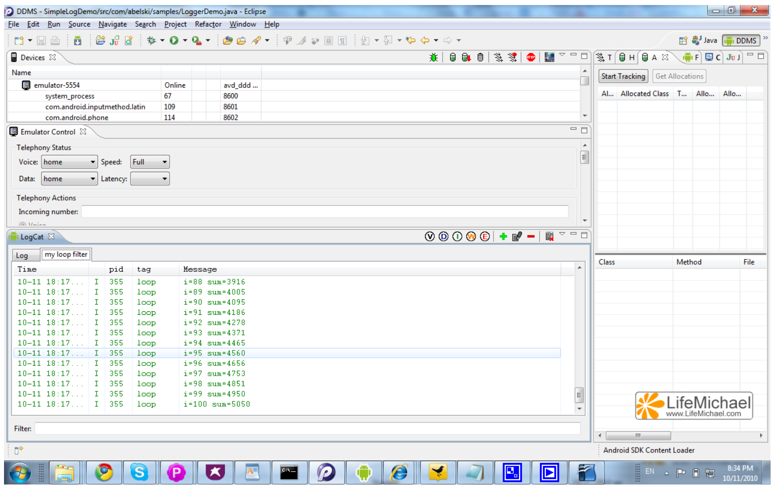
```
public class LoggerDemo extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button bt = (Button)findViewById(R.id.Button01);
        bt.setOnClickListener(new OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                int sum = 0;
                for(int i=1; i<=100; i++)
                {
                    sum += i;
                    Log.i("loop","i="+i+" sum="+sum);
                }
                EditText text = (EditText)findViewById(R.id.EditText01);
                text.setText("sum of 1..100 is "+sum);
            }
        });
    }
}
```

## Sample



```
[2010-10-11 20:11:28 - SimpleLogDemo] Installing SimpleLogDemo.apk...
[2010-10-11 20:11:53 - SimpleLogDemo] Success!
[2010-10-11 20:11:53 - SimpleLogDemo] Starting activity com.abelski.samples.LoggerDemo on device
[2010-10-11 20:11:58 - SimpleLogDemo] ActivityManager: Starting: Intent { act=android.intent.action
```

## Sample


DDMS - SimpleLogDemo/src/com/abelski/samples/LoggerDemo.java - Eclipse
File Edit Run Source Navigate Search Project Refactor Window Help
Java DDMS
Devices
Emulator Control
Start Tracking
Get Allocations
Class Method File
LifeMichael www.LifeMichael.com
Android SDK Content Loader

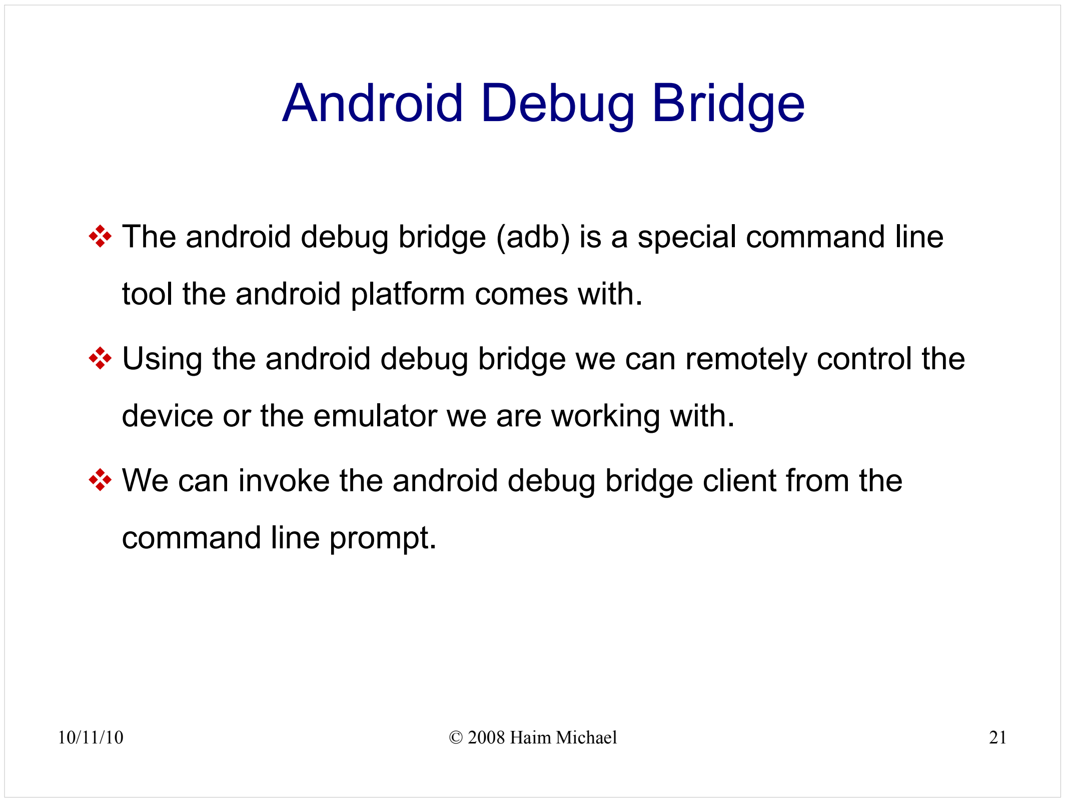
| Name | | | |
|---|---|---|---|
| emulator-5554 | Online | | avd_ddd ... |
| system_process | 67 | | 8600 |
| com.android.inputmethod.latin | 109 | | 8601 |
| com.android.phone | 114 | | 8602 |

Telephony Status
Voice: home Speed: Full
Data: home Latency:
Telephony Actions
Incoming number:

LogCat — Log | my loop filter

```
Time              pid  tag   Message
10-11 18:17...  I  355  loop  i=88 sum=3916
10-11 18:17...  I  355  loop  i=89 sum=4005
10-11 18:17...  I  355  loop  i=90 sum=4095
10-11 18:17...  I  355  loop  i=91 sum=4186
10-11 18:17...  I  355  loop  i=92 sum=4278
10-11 18:17...  I  355  loop  i=93 sum=4371
10-11 18:17...  I  355  loop  i=94 sum=4465
10-11 18:17...  I  355  loop  i=95 sum=4560
10-11 18:17...  I  355  loop  i=96 sum=4656
10-11 18:17...  I  355  loop  i=97 sum=4753
10-11 18:17...  I  355  loop  i=98 sum=4851
10-11 18:17...  I  355  loop  i=99 sum=4950
10-11 18:17...  I  355  loop  i=100 sum=5050
```

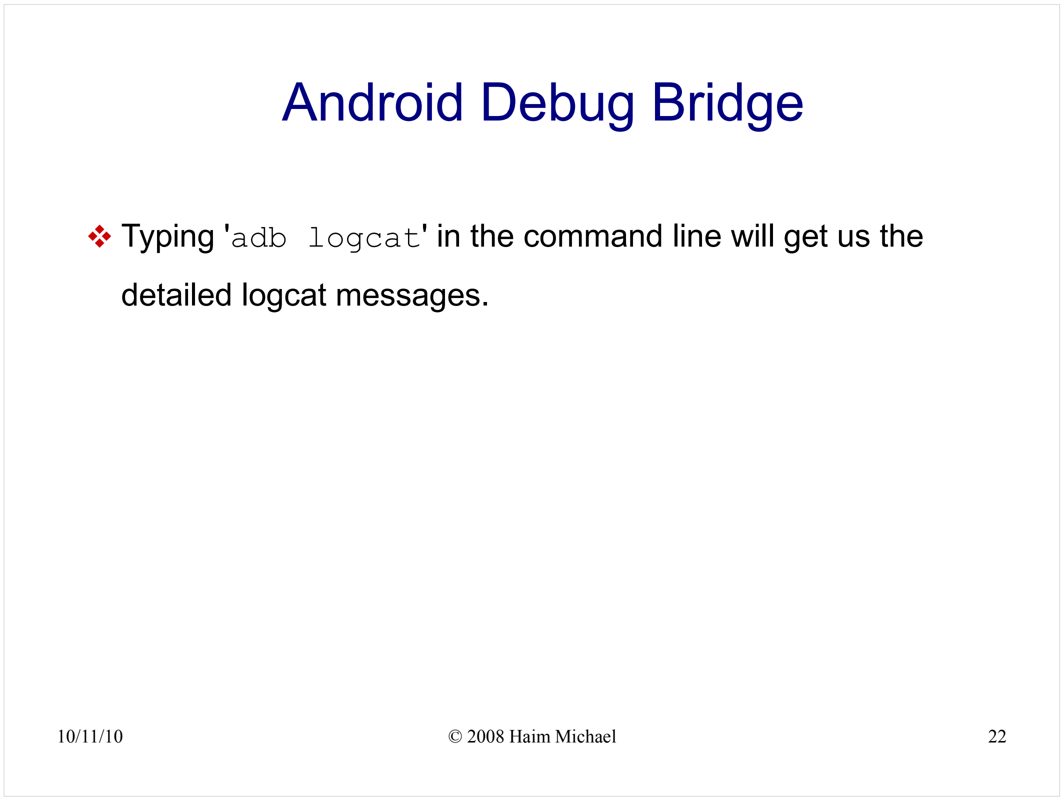
Filter:



## Android Debug Bridge

- The android debug bridge (adb) is a special command line tool the android platform comes with.
- Using the android debug bridge we can remotely control the device or the emulator we are working with.
- We can invoke the android debug bridge client from the command line prompt.

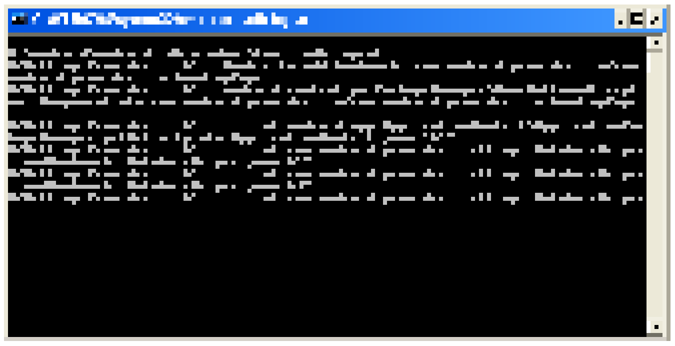
The server runs in the background. The server communicates either with the emulator or the handset it self. The communication itself is carried out using the TCP/IP protocol.



## Android Debug Bridge

- Typing '`adb logcat`' in the command line will get us the detailed logcat messages.

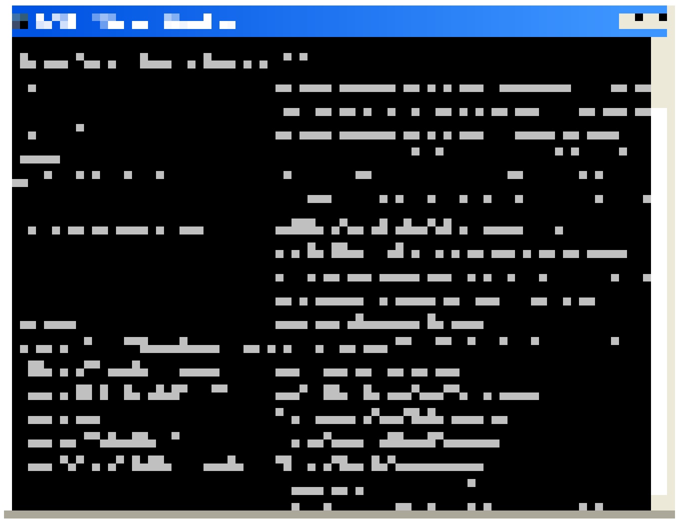
## Android Debug Bridge



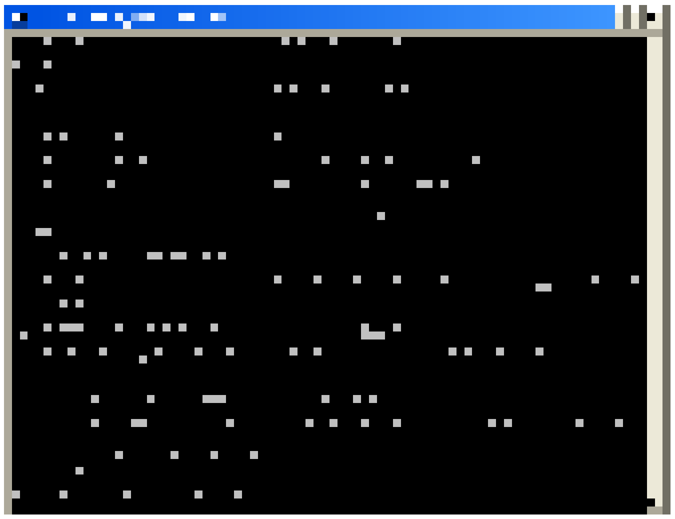
## Android Debug Bridge

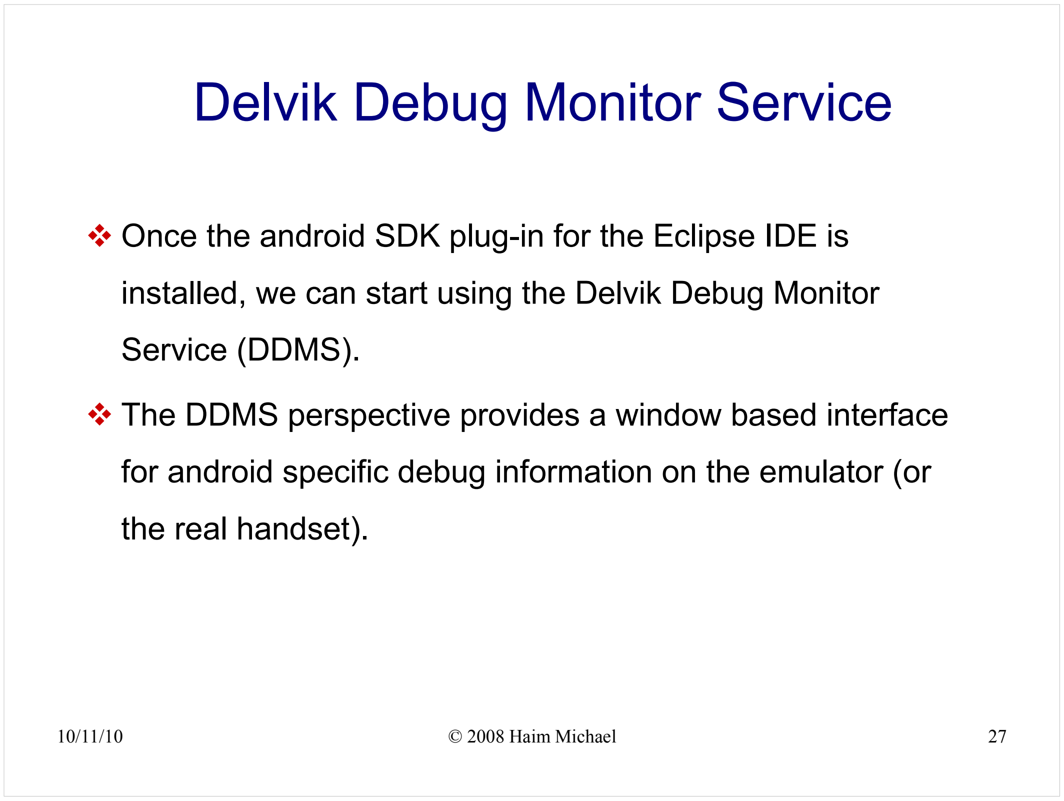
- Typing '`adb`' in the command line will get us a detailed list of all available `adb`'s commands.

## Android Debug Bridge



## Android Debug Bridge



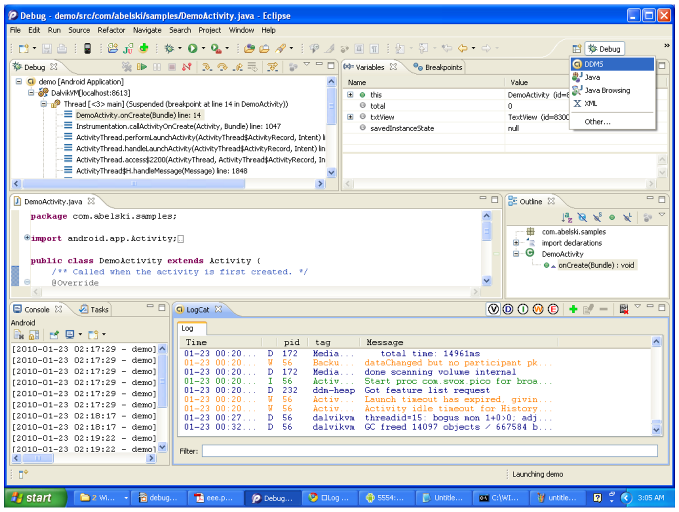


## Delvik Debug Monitor Service

- ❖ Once the android SDK plug-in for the Eclipse IDE is installed, we can start using the Delvik Debug Monitor Service (DDMS).
- ❖ The DDMS perspective provides a window based interface for android specific debug information on the emulator (or the real handset).

10/11/10 © 2008 Haim Michael 27

## Delvik Debug Monitor Service



```
demo [Android Application]
  DalvikVM[localhost:8613]
    Thread [<3> main] (Suspended (breakpoint at line 14 in DemoActivity))
      DemoActivity.onCreate(Bundle) line: 14
      Instrumentation.callActivityOnCreate(Activity, Bundle) line: 1047
      ActivityThread.performLaunchActivity(ActivityThread$ActivityRecord, Intent) li
      ActivityThread.handleLaunchActivity(ActivityThread$ActivityRecord, Intent) lin
      ActivityThread.access$2200(ActivityThread, ActivityThread$ActivityRecord, In
      ActivityThread$H.handleMessage(Message) line: 1848
```

| Name | Value |
|---|---|
| this | DemoActivity (id=8 |
| total | 0 |
| txtView | TextView (id=8300 |
| savedInstanceState | null |

```java
package com.abelski.samples;

import android.app.Activity;

public class DemoActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
```

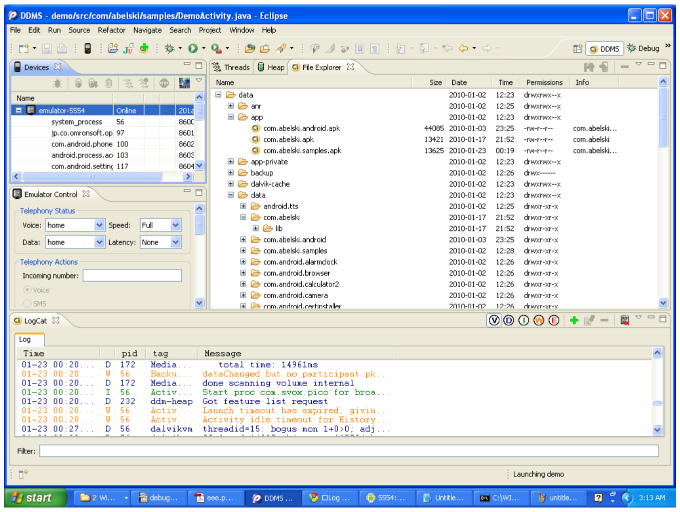
```
com.abelski.samples
import declarations
DemoActivity
  onCreate(Bundle) : void
```

```
[2010-01-23 02:17:29 - demo]
[2010-01-23 02:17:29 - demo]
[2010-01-23 02:17:29 - demo]
[2010-01-23 02:17:29 - demo]
[2010-01-23 02:17:29 - demo]
[2010-01-23 02:17:29 - demo]
[2010-01-23 02:18:17 - demo]
[2010-01-23 02:18:17 - demo]
[2010-01-23 02:19:22 - demo]
[2010-01-23 02:19:22 - demo]
```

```
Time              pid  tag       Message
01-23 00:20...  D  172  Media...     total time: 14961ms
01-23 00:20...  W  56   Backu...  dataChanged but no participant pk...
01-23 00:20...  D  172  Media...  done scanning volume internal
01-23 00:20...  I  56   Activ...  Start proc com.svox.pico for broa...
01-23 00:20...  D  232  ddm-heap  Got feature list request
01-23 00:20...  W  56   Activ...  Launch timeout has expired, givin...
01-23 00:20...  W  56   Activ...  Activity idle timeout for History...
01-23 00:27...  D  56   dalvikvm  threadid=15: bogus mon 1+0>0; adj...
01-23 00:32...  D  56   dalvikvm  GC freed 14097 objects / 667584 b...
```

10/11/10 © 2008 Haim Michael 28

## Delvik Debug Monitor Service



| Name | | | |
|---|---|---|---|
| emulator-5554 | Online | | 201a |
| system_process | 56 | | 8600 |
| jp.co.omronsoft.op | 97 | | 8601 |
| com.android.phone | 100 | | 8602 |
| android.process.ac | 103 | | 8603 |
| com.android.setting | 117 | | 8604 |

| Name | Size | Date | Time | Permissions | Info |
|---|---|---|---|---|---|
| data | | 2010-01-02 | 12:23 | drwxrwx--x | |
| anr | | 2010-01-02 | 12:25 | drwxrwx--x | |
| app | | 2010-01-02 | 12:23 | drwxrwx--x | |
| com.abelski.android.apk | 44085 | 2010-01-03 | 23:25 | -rw-r--r-- | com.abelski... |
| com.abelski.apk | 13421 | 2010-01-17 | 21:52 | -rw-r--r-- | com.abelski |
| com.abelski.samples.apk | 13625 | 2010-01-23 | 00:19 | -rw-r--r-- | com.abelski... |
| app-private | | 2010-01-02 | 12:23 | drwxrwx--x | |
| backup | | 2010-01-02 | 12:26 | drwx------ | |
| dalvik-cache | | 2010-01-02 | 12:23 | drwxrwx--x | |
| data | | 2010-01-02 | 12:23 | drwxrwx--x | |
| android.tts | | 2010-01-02 | 12:25 | drwxr-xr-x | |
| com.abelski | | 2010-01-17 | 21:52 | drwxr-xr-x | |
| lib | | 2010-01-17 | 21:52 | drwxr-xr-x | |
| com.abelski.android | | 2010-01-03 | 23:25 | drwxr-xr-x | |
| com.abelski.samples | | 2010-01-02 | 12:28 | drwxr-xr-x | |
| com.android.alarmclock | | 2010-01-02 | 12:26 | drwxr-xr-x | |
| com.android.browser | | 2010-01-02 | 12:26 | drwxr-xr-x | |
| com.android.calculator2 | | 2010-01-02 | 12:26 | drwxr-xr-x | |
| com.android.camera | | 2010-01-02 | 12:26 | drwxr-xr-x | |
| com.android.certinstaller | | 2010-01-02 | 12:26 | drwxr-xr-x | |

Telephony Status

Voice: home Speed: Full

Data: home Latency: None

Telephony Actions

Incoming number:

(•) Voice

( ) SMS

```
Time            pid  tag       Message
01-23 00:20...  D  172  Media...     total time: 14961ms
01-23 00:20...  W  56   Backu...  dataChanged but no participant pk...
01-23 00:20...  D  172  Media...  done scanning volume internal
01-23 00:20...  I  56   Activ...  Start proc com.svox.pico for broa...
01-23 00:20...  D  232  ddm-heap  Got feature list request
01-23 00:20...  W  56   Activ...  Launch timeout has expired, givin...
01-23 00:20...  W  56   Activ...  Activity idle timeout for History...
01-23 00:27...  D  56   dalvikvm  threadid=15: bogus mon 1+0>0; adj...
```

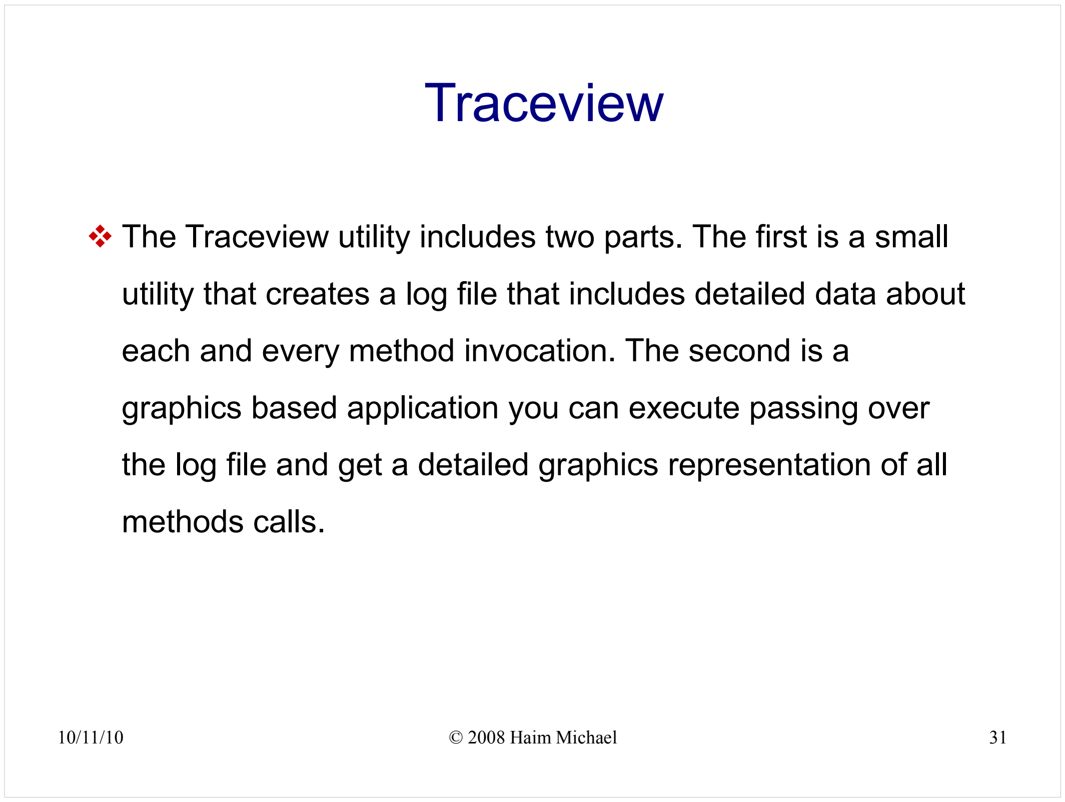
Filter:

Launching demo

10/11/10 © 2008 Haim Michael 29

## Traceview

- The Traceview utility allows tracking the exact methods that are been called as well as the exact time each one of these methods executions takes.



## Traceview

- The Traceview utility includes two parts. The first is a small utility that creates a log file that includes detailed data about each and every method invocation. The second is a graphics based application you can execute passing over the log file and get a detailed graphics representation of all methods calls.

## Traceview

- In order to use the Traceview utility we should first import the `android.os` package and add to our code the following lines.

```
...
Debug.startMethodTracing(“mytrace”);
...

...
Debug.stopMethodTracing();
...
```

## Traceview

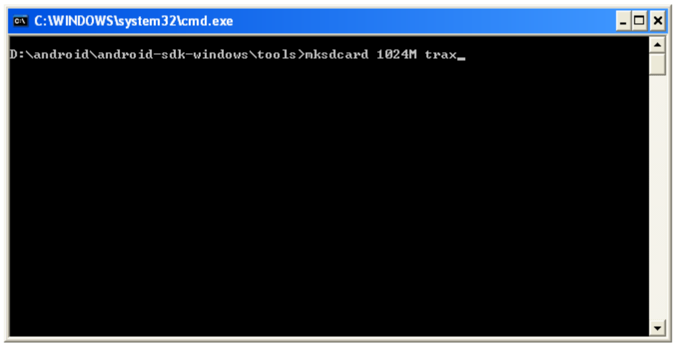
- The Tracebiew utility will fill in the log file on the android SD card with data generated during the code execution.
- If we work with the emulator then we should create an AVD with a virtual SD card.

## Traceview

- The first step should be creating the file on the SD card:

```
mksdcard 10M traco
```

## Traceview

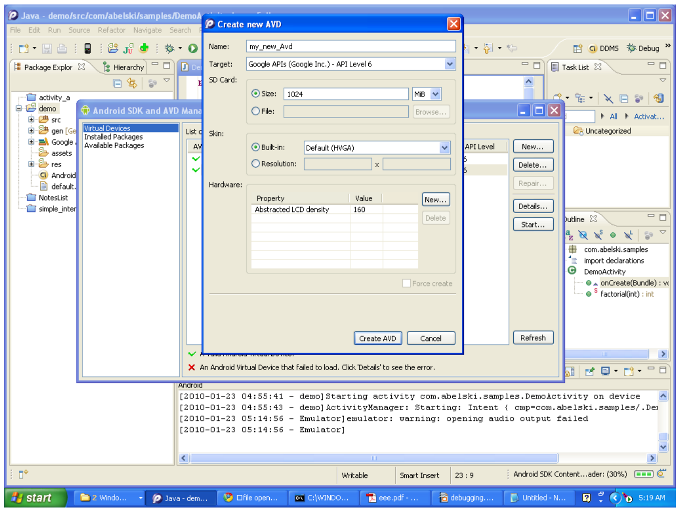


```
D:\android\android-sdk-windows\tools>mksdcard 1024M trax_
```

## Traceview

- The second step should be creating the AVD we want to use specifying the size of the requested SD card.

## Traceview



Create new AVD

Name: my_new_Avd

Target: Google APIs (Google Inc.) - API Level 6

SD Card: (•) Size: 1024 MiB; ( ) File:

Skin: (•) Built-in: Default (HVGA); ( ) Resolution:

Hardware:

| Property | Value |
|---|---|
| Abstracted LCD density | 160 |

☐ Force create

Create AVD | Cancel

An Android Virtual Device that failed to load. Click 'Details' to see the error.

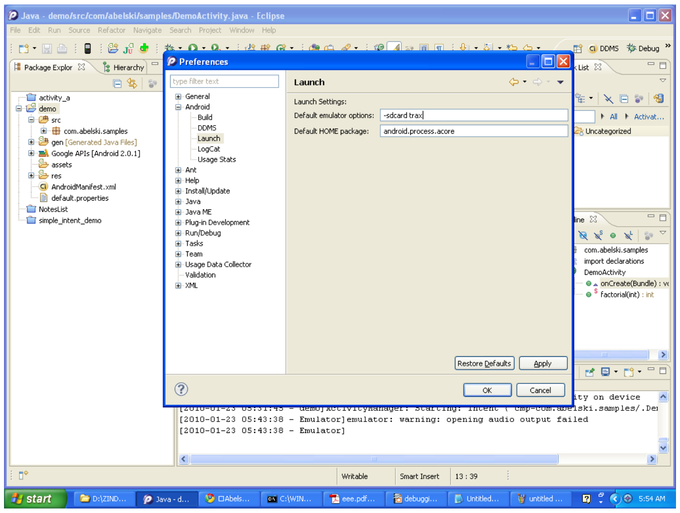
```
Android
[2010-01-23 04:55:41 - demo]Starting activity com.abelski.samples.DemoActivity on device
[2010-01-23 04:55:43 - demo]ActivityManager: Starting: Intent { cmp=com.abelski.samples/.De
[2010-01-23 05:14:56 - Emulator]emulator: warning: opening audio output failed
[2010-01-23 05:14:56 - Emulator]
```

Writable | Smart Insert | 23 : 9 | Android SDK Content...ader: (30%)

## Traceview

- The third step should be telling the emulator we want to use a virtual SC card. In the Eclipse IDE you should choose `Window > Preferences > Android > Launch`. Within the box for the emulator options you should add the following code `-sdcard ./traco`
- Make sure to specify the complete path to the file, so the emulator can always find it.

## Traceview



Launch

Launch Settings:

Default emulator options: -sdcard trax

Default HOME package: android.process.acore

```
[2010-01-23 05:43:38 - Emulator]emulator: warning: opening audio output failed
[2010-01-23 05:43:38 - Emulator]
```

## Traceview

- The fourth step would be executing our code. Our code should include the call to start tracing the methods.

```
Debug.startMethodTracing("traco"); //traco is the filename
```

  In addition, our code should include a call to stop it.

```
Debug.stopMethodTracing();
```

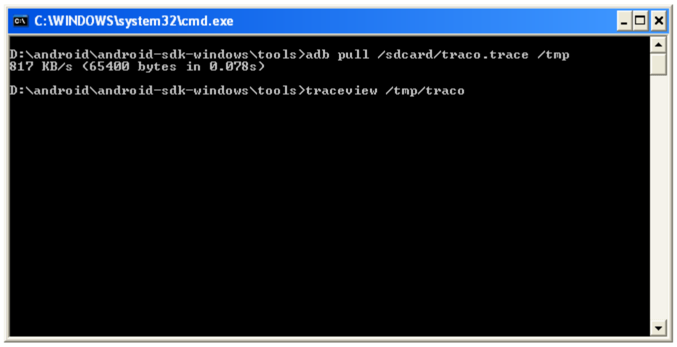
- When the application calls `startMethodTracing()`, the system creates a file called ___.trace (e.g. traco.trace). This file contains the binary method trace data and its mapping table with thread and method names.

## Traceview

- When using the Traceview utility the execution times are significantly slower. Therefore, we shouldn't refer these times as the accurate one. We can only compare them with each other.
- Once the execution completes we can get the log data displayed in a graphics way.

```
adb pull /sdcard/traco.trace /tmp
traceview /tmp/traco
```

## Traceview

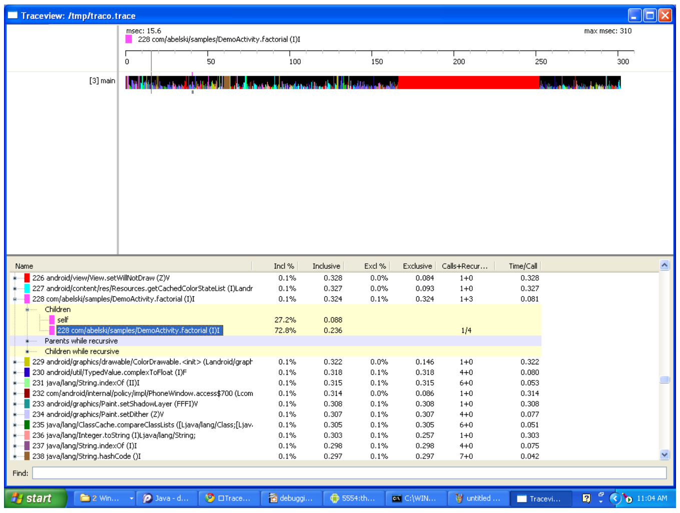


C:\WINDOWS\system32\cmd.exe

```
D:\android\android-sdk-windows\tools>adb pull /sdcard/traco.trace /tmp
817 KB/s (65400 bytes in 0.078s)

D:\android\android-sdk-windows\tools>traceview /tmp/traco
```

10/11/10 © 2008 Haim Michael 42

## Traceview



228 com/abelski/samples/DemoActivity.factorial (I)I

| Name | Incl % | Inclusive | Excl % | Exclusive | Calls+Recur… | Time/Call |
|---|---|---|---|---|---|---|
| 226 android/view/View.setWillNotDraw (Z)V | 0.1% | 0.328 | 0.0% | 0.084 | 1+0 | 0.328 |
| 227 android/content/res/Resources.getCachedColorStateList (I)Landr | 0.1% | 0.327 | 0.0% | 0.093 | 1+0 | 0.327 |
| 228 com/abelski/samples/DemoActivity.factorial (I)I | 0.1% | 0.324 | 0.1% | 0.324 | 1+3 | 0.081 |
| Children | | | | | | |
| self | 27.2% | 0.088 | | | | |
| 228 com/abelski/samples/DemoActivity.factorial (I)I | 72.8% | 0.236 | | | 1/4 | |
| Parents while recursive | | | | | | |
| Children while recursive | | | | | | |
| 229 android/graphics/drawable/ColorDrawable.<init> (Landroid/graph | 0.1% | 0.322 | 0.0% | 0.146 | 1+0 | 0.322 |
| 230 android/util/TypedValue.complexToFloat (I)F | 0.1% | 0.318 | 0.1% | 0.318 | 4+0 | 0.080 |
| 231 java/lang/String.indexOf (II)I | 0.1% | 0.315 | 0.1% | 0.315 | 6+0 | 0.053 |
| 232 com/android/internal/policy/impl/PhoneWindow.access$700 (Lcom | 0.1% | 0.314 | 0.0% | 0.086 | 1+0 | 0.314 |
| 233 android/graphics/Paint.setShadowLayer (FFFI)V | 0.1% | 0.308 | 0.1% | 0.308 | 1+0 | 0.308 |
| 234 android/graphics/Paint.setDither (Z)V | 0.1% | 0.307 | 0.1% | 0.307 | 4+0 | 0.077 |
| 235 java/lang/ClassCache.compareClassLists ([Ljava/lang/Class;[Ljav. | 0.1% | 0.305 | 0.1% | 0.305 | 6+0 | 0.051 |
| 236 java/lang/Integer.toString (I)Ljava/lang/String; | 0.1% | 0.303 | 0.1% | 0.257 | 1+0 | 0.303 |
| 237 java/lang/String.indexOf (I)I | 0.1% | 0.298 | 0.1% | 0.298 | 4+0 | 0.075 |
| 238 java/lang/String.hashCode ()I | 0.1% | 0.297 | 0.1% | 0.297 | 7+0 | 0.042 |