

# Android RunTime

# Introduction

- ❖ Till Android 5.0, the Dalvik VM was the android execution runtime environment. As of Android 5.0, there is a new execution runtime environment. Its name is Art.

# Introduction to Dalvik VM

- ❖ The Dalvik VM is a virtual machine, designed and written by Dan Bornstein with the help of more than a few software engineers from Google.
- ❖ The Dalvik VM was developed as part of the Android mobile phone platform.
- ❖ The Dalvik VM is similar to the JVM or .NET CLR.

# Introduction to Dalvik VM

- ❖ The Dalvik VM is optimized for low memory requirements, and is capable of allowing multiple VM instances to run at the same time.



# Dalvik VM is not a JVM

- ❖ The Dalvik VM is not a JVM. The bytes the Dalvik VM works on are not Java byte code.

# Dalvik VM `dex` File Format

- ❖ The `.dex` (Dalvik Executable) files are zipped into a single `.apk` file.
- ❖ The Dalvik VM is capable of running files of this format only.

# Dalvik VM `dx` Tool

- ❖ The `dx` tool is one of the tools the Android SDK includes.
- ❖ This tool transforms Java `.class` files into the `.dex` format.

# Dalvik VM Garbage Collector

- ❖ The Dalvik VM includes a garbage collector that functions similarly to the Java VM garbage collector.



# Dalvik VM Garbage Collector

- ❖ We can access the shell command of the Linux operating system installed on the android handset (`adb shell`) and call the `setprop` utility in order to set specific key-value pairs... such as those that will effect the way the Dalvik VM works.

```
adb shell setprop <name> <value>
```

# Dalvik VM Garbage Collector

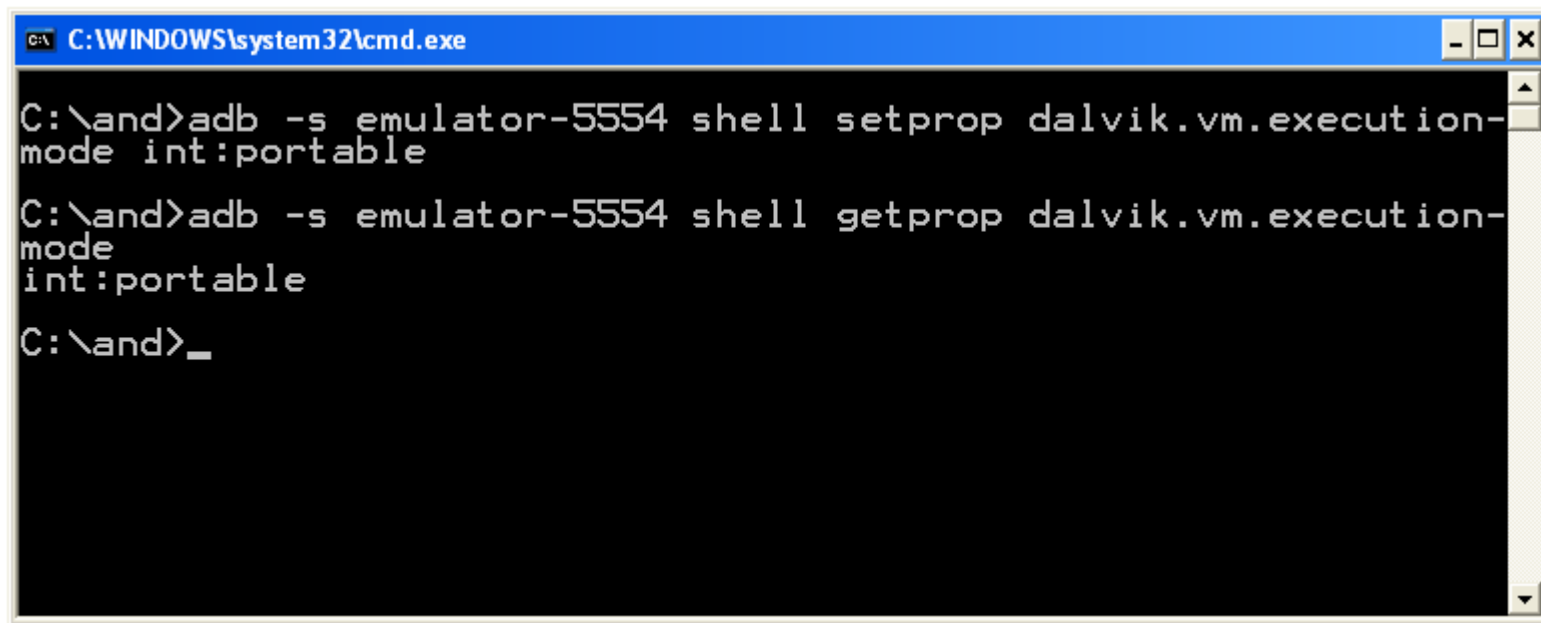
- ❖ Calling this command, we must restart the android runtime execution environment. Otherwise, the changes won't take effect (`adb shell stop; adb shell start`).

# Dalvik VM Garbage Collector

- ❖ We can access the shell command of the Linux operating system installed on the android handset (`adb shell`) and call the `getprop` utility in order to get the value of a specific key.

```
adb shell getprop <name>
```

# Dalvik VM Garbage Collector



```
C:\WINDOWS\system32\cmd.exe

C:\>adb -s emulator-5554 shell setprop dalvik.vm.execution-
mode int:portable

C:\>adb -s emulator-5554 shell getprop dalvik.vm.execution-
mode
int:portable

C:\>_
```

# Introduction to Art VM

- ❖ Art is a new improved running environment that was developed for the Android platform.
- ❖ Most of applications that were developed for Dalvik VM should work on Art without any problem.
- ❖ Art is capable of executing dex files. The same format been used by Dalvik.

# Art's Ahead-of-time (AOT) compilation

- ❖ When a new application is installed, Art compiles the application's `dex` files into `oat` ones. The compilation is performed using the `dex2oat` utility.
- ❖ This ahead of time compilation improves the application performance.

# Art's Improved Garbage Collection

- ❖ Art introduces an improved garbage collector. This improved garbage collector might be highly important when dealing with applications with bad performance.

# Art's Profiler

- ❖ Art introduces a new better sampling profiler with more capabilities. Unlike traceview, Art's profiler results are not influenced by the per-method-call overhead.



# Art's Debugging Capabilities

- ❖ Art allows us to debug our code in a better way. One of the debugging improvements allows us to get the exact number of instances that were created from a specific class.

# Improved Crash Reports

- ❖ When the application crashes, the crash reports Art provides us with through its log messages, include more information.