



Developing for Android with Eclipse



EclipseDay at Googleplex 2009
August 27th, 2009

Xavier Ducrohet - Google Inc.



Why custom plug-ins?



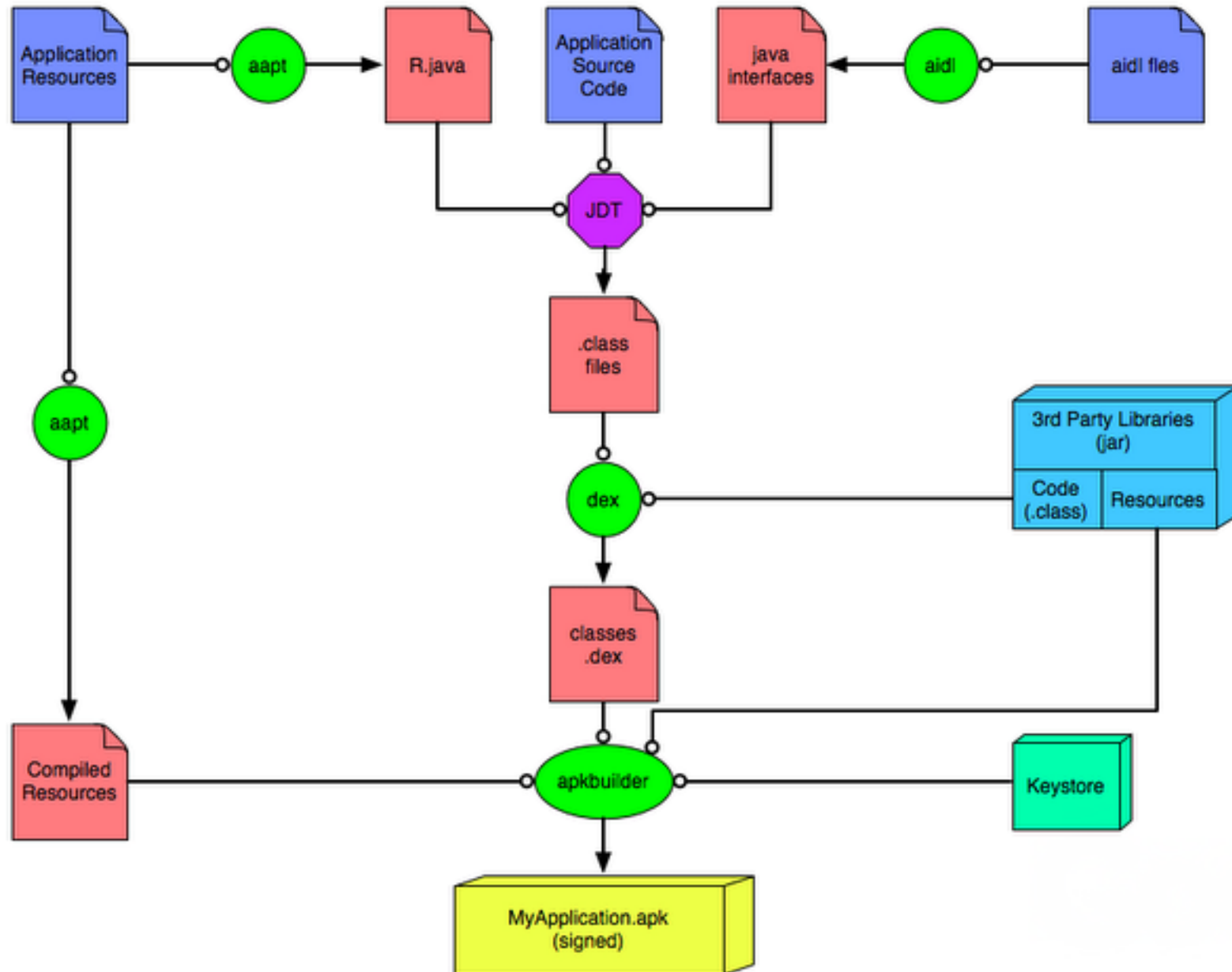
- Android build is complex
- Setting up debugger is non-trivial
- Lots of external tools

Goals of the plug-ins?

- Provide familiar work flow
- Hide all the android-specific stuff



Build Process



Android Projects



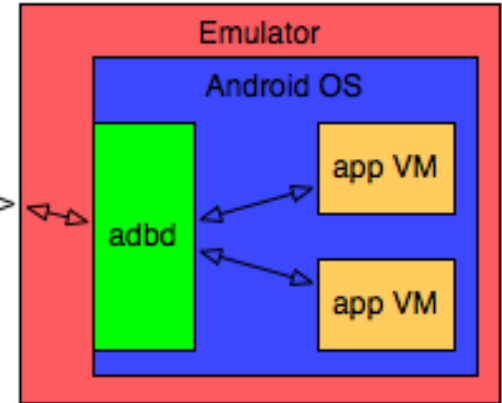
- Project natures
 - Java (JDT)
 - Android
- Two IncrementalProjectBuilder
 - PreCompiler
 - Resources -> R.java (*aapt*)
 - Compile aidl files (*aidl*)
 - Package builder
 - Convert to Dalvik bytecode (*dx.jar*)
 - Compile resources into binary XMLs (*aapt*)
 - Package everything into APK
 - Sign with debug key

Debugger - DDMS - Emulator



DDMS

Name			
emulator-tcp-5555	Online		xav
system_process	460		8600
com.google.android.phone	557		8601
com.google.android.home	561		8602
com.test.hello	659		8610

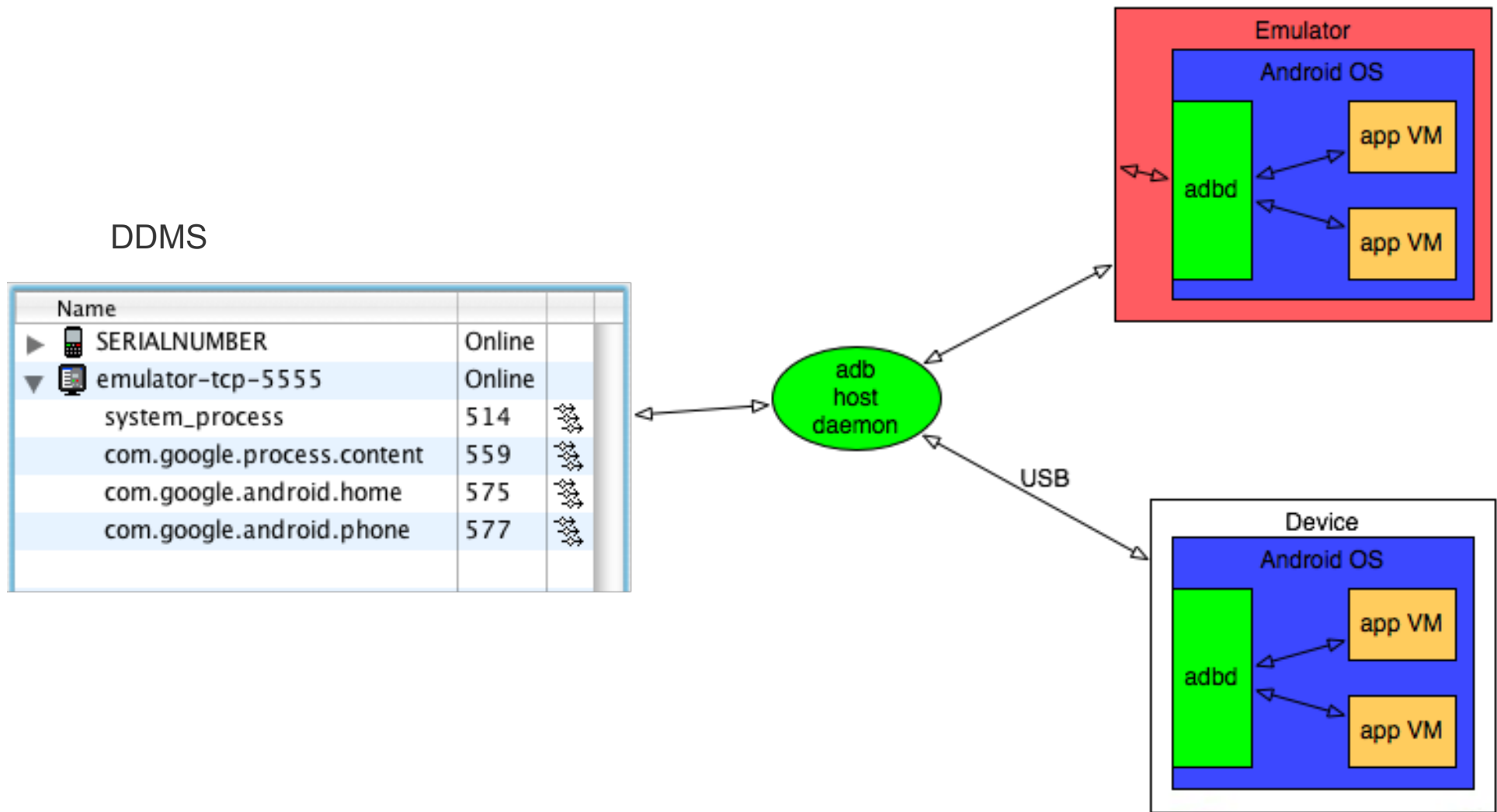


Debug window showing the execution of a Hello application. The thread is suspended at line 21 in MyActivity.onCreate(Bundle).

```
Debug [x]
Hello [Android Application]
  DalvikVM[localhost:8610]
    Thread [<3> main] (Suspended (breakpoint at line 21 in MyActivity))
      MyActivity.onCreate(Bundle) line: 21
      Instrumentation.callActivityOnCreate(Activity, Bundle) line: 1021
      ActivityThread.performLaunchActivity(ActivityThread$ActivityRecord)
      ActivityThread.handleLaunchActivity(ActivityThread$ActivityRecord)
      ActivityThread.access$2000(ActivityThread, ActivityThread$ActivityRecord)
      ActivityThread$ActivityRecord(ActivityThread$Handler).handleMessage(Message)
      ActivityThread$Handler.dispatchMessage(Message) line: 89
      Looper.loop() line: 126
```



Plugin - Device communication



Debugging on Devices



- Secure device => Cannot debug any app
- debuggable = true in manifest to enable debugger
- Don't ship with this!

DDMS



- Handle connection to devices through adb
- Basic tools
 - device/app list
 - logcat
 - heap / thread views
 - emulator control
 - Screen capture
 - File Explorer

Editing Android files



- Java is handled by JDT
- XML files
 - Android Manifest
 - Values (strings, colors, ...)
 - Layouts
 - Menu definition
 - Settings definition
- 9-patch bitmaps
 - not yet integrated into ADT

XML Editors



- "Advanced"
 - Form based
 - WYSIWYG
- Text Editor
 - Default XML text editor
 - Custom content assist
- Resource Manager
 - Load each project resources
 - customize editors with project content (content assist)
 - Resource Explorer
- Refactoring
 - Extract Strings

Layout Editor



- Lots of challenges
 - Rendering fidelity
 - Complex user interactivity
 - UI for a lot of attributes
- Current version
 - Rendering
 - Property View for attributes
 - *Very* basic drag and drop

Layout Rendering: Architecture



Layout Rendering: *layoutlib*



- Library bundled with the SDK
 - 100% Java
 - Loaded dynamically by ADT
 - Stateless
- Android View System
- 2D Drawing API reimplemented on top of Java2D
- Resource Manager API used by View System
 - Query Resources
 - Resolve Theme/reference
- Resources parsed by ADT



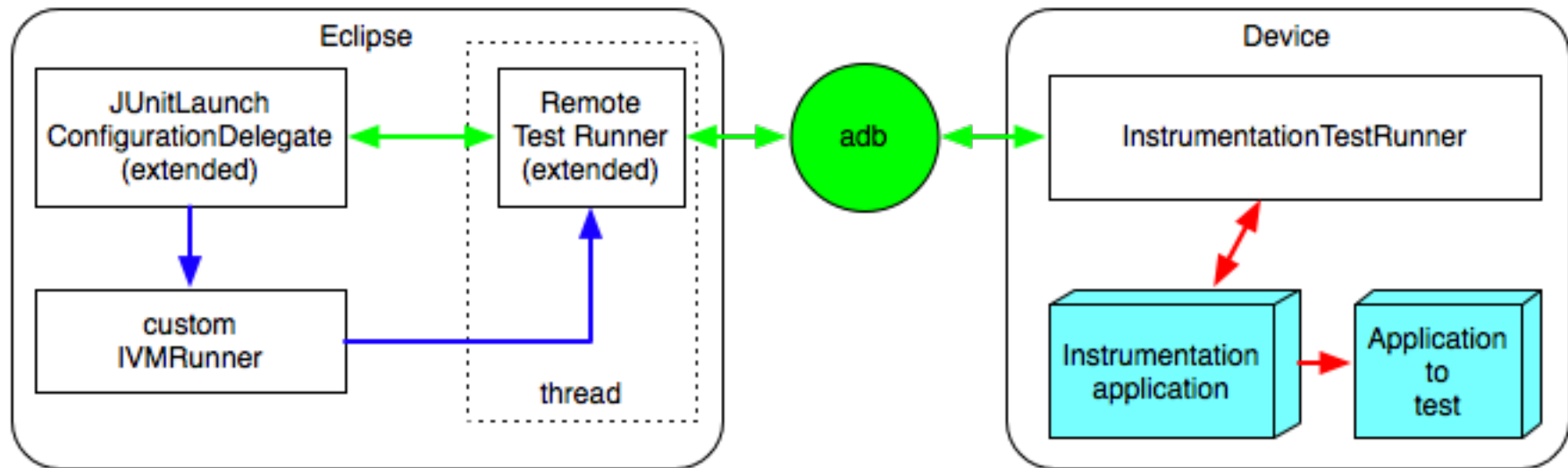
Testing



- *android.jar* has no code
 - Cannot run tests on the desktop JVM
- Android Instrumentation Framework
 - Runs JUnit tests on the device
 - Basic command line:

```
adb shell am instrument ...
```
 - Text output
- Integration into Eclipse

Running JUnit Tests



Profiling



- TraceView
 - Standalone Tool (SWT) to see traces
- hprof files
 - Non standard, but converter available

Useful Links



- <http://developer.android.com>
 - SDK / ADT download
 - Dev Guide, API reference
 - Developer mailing lists
- <http://source.android.com>
 - Android source code
 - Dev Tools source code
 - Platform mailing lists

