Multiwindow support on Android

Andrzej Wieczorek    Mikel Echegoyen

Tieto Product Development Services
About us…

• Andrzej Wieczorek
  • Business Development Manager
  • responsible for Android product creation and connectivity areas for mobile and embedded devices

• Mikel Echegoyen (@echegmik)
  • Business Development Director, Semiconductors
  • 14 years in Mobile R&D on multiple roles from development to sales

• Tieto Product Development Services
  • R&D in communications and embedded technologies
  • Part of Tieto, 14000 employees, headquarters in Finland
  • More at www.tieto.com/pds
60 second demo
=
a thousand words
Déjà Vu, Much?

Cornerstone

Rockchip

Samsung

Omnitron

Sony

Multitasking Apps

and more…
Don’t reinvent the wheel… rather inflate & add more!

Cornerstone ICS (4.0/4.1)

“Cornerstone Port to 4.2.2”

Add float mode, Settings, features

Re-architect for KitKat 4.4

“It’s dead, Jim”

“Running on latest”

“Racing ready”

“Get new tires!”

“came back to life!”
Multiwindow feature overview

Floating windows

- Focus, Move, Resize, Close Window
- Windows Manager application
  - Show/hide/edit windows (e.g. resize)
  - Move Window Manager (left, right)
  - Toggle floating/docked windows UX
  - Add new Window
  - Add new tab (group windows)

Docked windows

- Move window to Home Screen area, Close
- Docked Windows Area
- Home Screen area
## Feature comparison

<table>
<thead>
<tr>
<th>Feature</th>
<th>Tieto MW</th>
<th>Cornerst.</th>
<th>Rockchip</th>
<th>Ixonos</th>
<th>Samsung</th>
<th>Sony</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic features: open, close, move, resize, maximize, etc.</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Advanced features: group windows, swap</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Docked windows UX</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Floating windows UX</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>MultiInstance for Apps</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>CTS</td>
<td>✗</td>
<td>✗</td>
<td>?</td>
<td>?</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>OpenSource</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Transparent for apps</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>?</td>
<td>✗ Flag req.</td>
<td>✗</td>
</tr>
<tr>
<td>Android Version</td>
<td>4.2.2, 4.4</td>
<td>4.0, 4.1, 4.3</td>
<td>4.2.2</td>
<td>4.2.2</td>
<td>4.2+</td>
<td>4.0+</td>
</tr>
</tbody>
</table>
Architecture changes for multiwindow

Applications

- Home
- Contacts
- Phone
- Browser
- ...

Application Framework

- Activity Manager
- Window Manager
- View Systems
- Location Manager
- Package Manager
- Telephony Manager
- Content Providers
- Resources Manager
- Notification Manager

Libraries

- Android Runtime

Linux Kernel
Multiple Activity stacks

Activity stack

Main Stack (Home)
Cornerstone Stack
Panel Stack 1
Panel Stack 2

Cornerstone

Main Stack (Home)
Window Manager Stack
Activity Stack
Activity Stack

Tieto
How it works

**Multiwindow Application**
- Start activity
- Manage windows

**ActivityManagerService**
- Start activity management
- Stacks management
- Resume state

**PhoneWindow**
- Add decors for swap, maximize, close, resize
- Manage windows

**WindowManagerService**
- Window positions management
- Activity – window relationship
- Windows state management
ActivityManagerService changes
Jelly Bean (4.2)

Add 2 classes introducing multiple activity stacks

int initWindow(Rect position) - creates new stack and window
void relayoutWindow(int stackId, Rect position) - changes position
void removeWindow(int stackId) - remove window from a screen

ActivityRecord
ActivityStack
StackAdapter
ActivityManagerService
void startCornerstoneApp(intent, stackId);
StackAdapterContainer
ActivityManagerService changes

KitKat

---

Changes as in JB

Removing stack adapters since KitKat introduces the supervisor
WindowManagerService changes

Jelly Bean (4.2)

Created by Cornerstone
Tieto modified to have multiple panels

WindowPanel

WindowManagerService

Changes to have floating windows, Z-order and manage inputs

DisplayContent

WindowList
WindowManagerService changes

KitKat

„Migration” from WindowPanel to StackBox introduced in KitKat

As in JB

Removed „only 2 stack boxes” constraint
Use cases
New activity, KitKat

Launcher

```java
intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_RUN_IN_WINDOW);
startActivity(app.intent);
```

StackSupervisor

```java
int stackId =
mService.createStack(-1, parentStackId, isMultiwindow ? StackBox.TASK_FLOATING : StackBox.TASK_STACK_GOES_OVER, 1.0f);
```

WindowManagerService

```java
StackBox newBox = new StackBox(mService, this, position, null);
```
Use cases

Resize or move, KitKat

Active window sends relayout command to AMS

AMS passes command to WMS

WMS updates StackBox's position or size

WMS recalculates all widows positions or sizes

PhoneWindow

ActivityManagerNative.getDefault().relayoutWindow(getStackBoxId(), mNewFrame);

ActivityManagerService

mWindowManager.relayoutWindow(stackID, r);

WindowManagerService

for (StackBox sb : mStackBoxes) {
    if ((sb.getStackId() == stackBoxId) &&
         (sb.relayoutStackBox(pos))) {
        layoutNeeded = true;
        return true;
    }
}

WindowManagerService

performLayoutAndPlaceSurfacesLocked()
Multiwindow support in KitKat

Status

- AOSP re-design work ongoing
- Visible today:
  - ActivityStackSupervisor
  - StackBoxes
  - Screen split functionality – early stage
  - Move windows, resize
  - New activity manager API (not public for apps)

```
adb shell> am stack boxes
...
adb shell> am stack create <task_id> <rel_stackbox_id> <position> <weight>
```

At this stage SDK level accesses to the classes would enable developers to create their own „multiwindow layout managers”
Lessons learned

User experience

• Reload config (layout resources) files when resizing? How often?
  • Yes → possible flickering
  • No → graphics not adjusted to a size

• Many apps don’t look perfect when resized, duh!
  • No wonder Samsung has a whitelist

• Hard to get back when experiencing multiwindow 😊
Lessons learned

Resources and performance

- Small performance penalty [1-3%]
  - About 20 apps running when OOM killer activates
  - CPUs don’t break a sweat (Dual core, Quad Core, ARM & X86)
- Redrawing complex apps is slow, option is to show only borders when resizing
- Potential conflicts when accessing resources
- Multi-Instance behavior is different across applications
Lessons learned

New opportunities

Social TV
„Watch and chat”

Dual screen, dual OS
„Debian on Android”

Remote control
Using Multiwindow API
“Talk is cheap, show me the code”

- [https://github.com/tieto/multiwindow_for_android/tree/tieto_multiwindow](https://github.com/tieto/multiwindow_for_android/tree/tieto_multiwindow) (4.2.2)
- [https://github.com/tieto/multiwindow_for_android](https://github.com/tieto/multiwindow_for_android) (4.4)
- All needed code as well instructions to build and run available
- Licenses:
  - [Apache 2.0](https://github.com/tieto/multiwindow_for_android/tree/tieto_multiwindow) (frameworks, AOSP apps)
  - [GPLv3](https://github.com/tieto/multiwindow_for_android) (Tieto reference apps)
## Git activity summary *

<table>
<thead>
<tr>
<th>Package</th>
<th>Changes</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>frameworks/base</td>
<td>39 files changed, 9443 insertions(+), 133 deletions(-)</td>
<td>Cornerstone</td>
</tr>
<tr>
<td>frameworks/base</td>
<td>51 files changed, 2587 insertions(+), 5369 deletions(-)</td>
<td></td>
</tr>
<tr>
<td>packages/apps/ TietoLauncher</td>
<td>20 files changed, 947 insertions(+)</td>
<td>Tieto</td>
</tr>
<tr>
<td>packages/apps/ TietoMultiWindow</td>
<td>50 files changed, 2658 insertions(+)</td>
<td></td>
</tr>
<tr>
<td>packages/apps/Launcher2</td>
<td>4 files changed, 30 insertions(+), 20 deletions(-)</td>
<td></td>
</tr>
<tr>
<td>packages/apps/Settings</td>
<td>9 files changed, 344 insertions(+)</td>
<td></td>
</tr>
</tbody>
</table>

* Numbers for Jelly Bean  
For KitKat approx. 1000 lines changed
# Project FAQ

<table>
<thead>
<tr>
<th>Questions</th>
<th>Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>How to build &amp; use it?</td>
<td>Instructions in readme and project wiki</td>
</tr>
<tr>
<td>Where is my ROM?</td>
<td>We don’t provide ROMS, sorry. You can build your own 😊</td>
</tr>
<tr>
<td>Can I contribute?</td>
<td>Sure, pull requests welcomed.</td>
</tr>
<tr>
<td>How do I get support?</td>
<td>Create github “issues” in Multiwindow project. We’ll do our best.</td>
</tr>
<tr>
<td>Will you support Device X?</td>
<td>The solution is HW agnostic (device, architecture). We test in a wide numbers of devices (nexus, Xperia, x86 bay trail, ..)</td>
</tr>
<tr>
<td>Is the solution CTS compliant?</td>
<td>Not at this time</td>
</tr>
</tbody>
</table>
What next?

- Follow up upstream 4.4 and next
- New Launcher
- Optimizations
- CTS compliancy
- Collaborate with you 😊
That was all folks, Thanks!

Questions?

Andrzej Wieczorek
andrzej.wieczorek@tieto.com

Mikel Echegoyen
mikel.echegoyen@tieto.com