



# Building Android Apps for a Global Audience

Yiwen Zhan  
Product Manager, Internationalization

Fabrice Di Meglio  
Software Engineer, Android Framework

# Android has global momentum

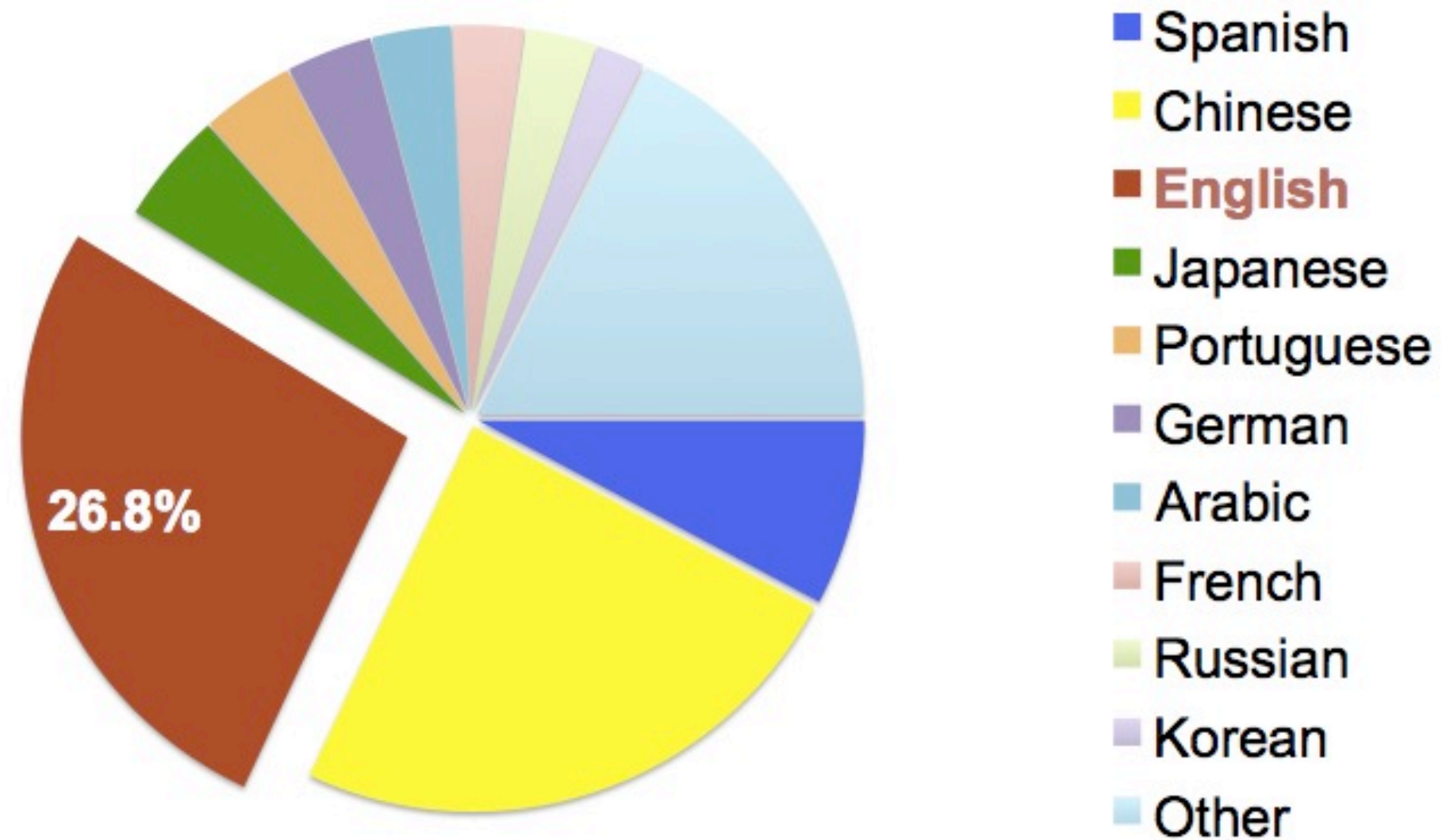
More than **900 million** Android devices worldwide

**1.5 million** new Android devices every day

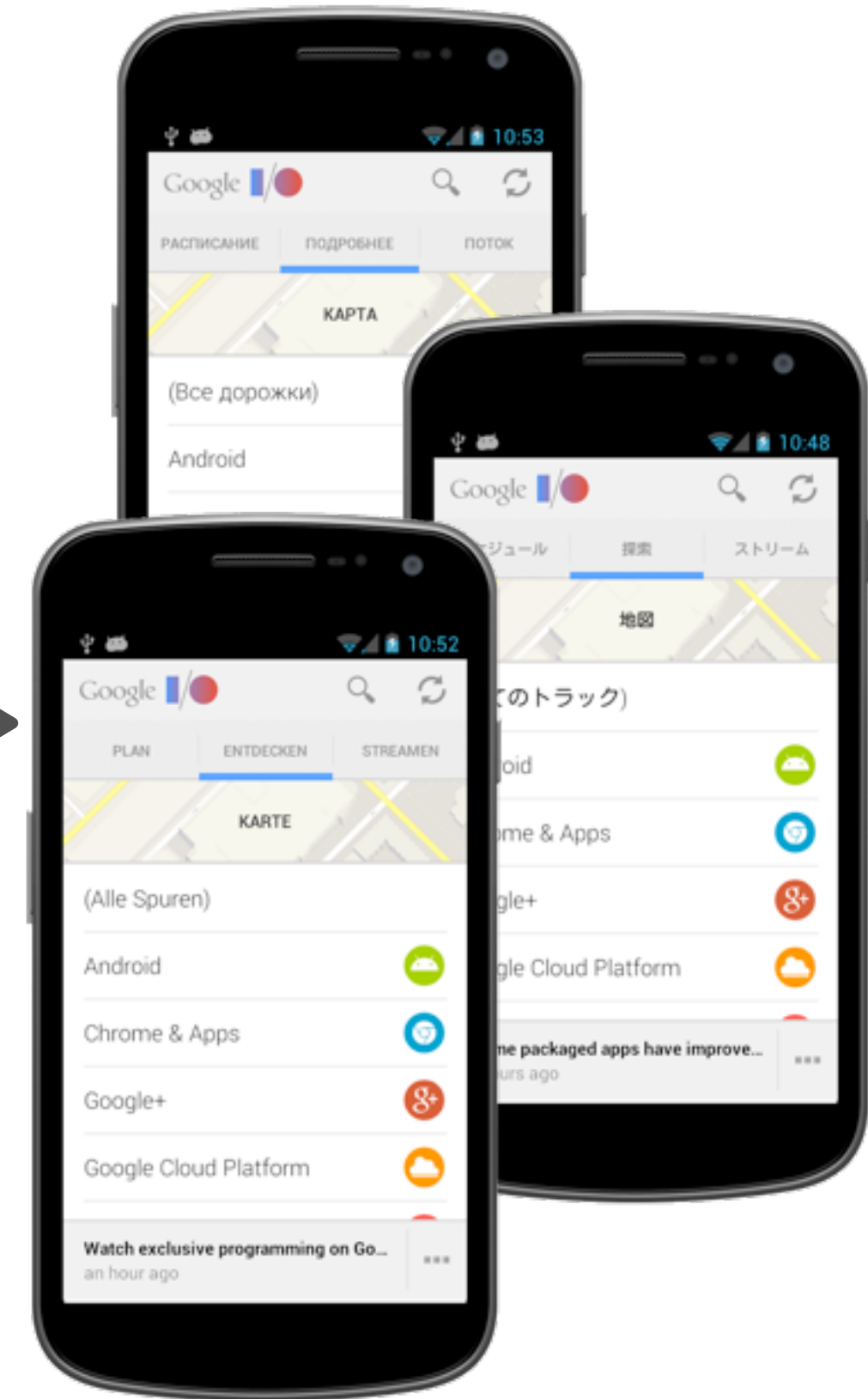
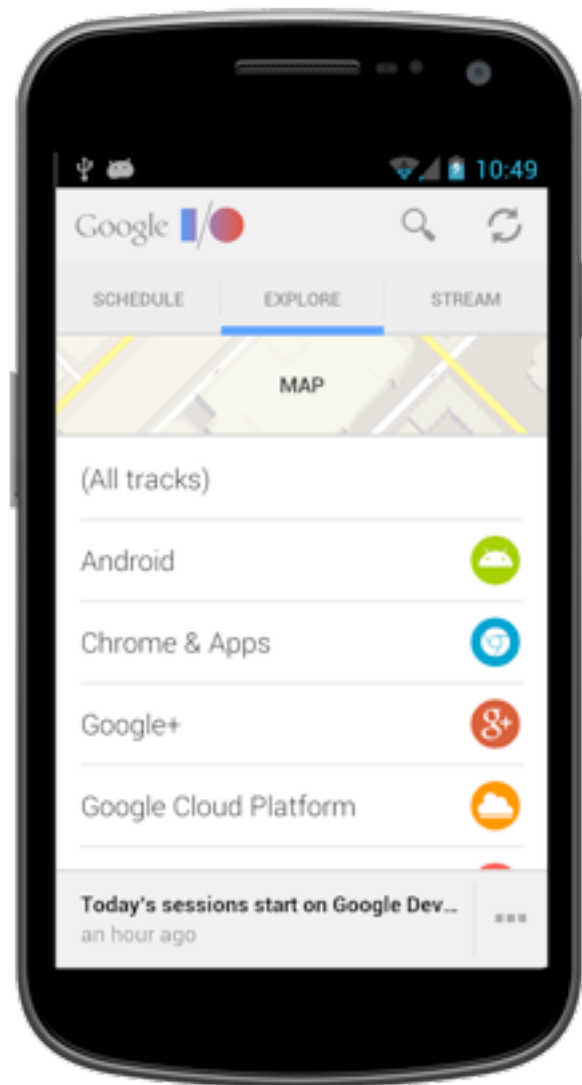
**48 billion** apps downloaded in Play from **190 countries**



# Internet users by language



# What does it take?



# A strategy for language expansion

## Metrics

- Top languages used on the web
- Top countries by mobile phone usage
- Top countries by paid apps installs
- Metrics specific to app category

## Language Groups

- Latin script (Spanish, French)
- Non-Latin European (Greek, Russian)
- CJK (Chinese, Japanese, Korean)
- SSEA (Thai, Hindi)
- RTL (Arabic, Hebrew, Persian)



# Internationalization and Localization

**Internationalization - i18n**

**Design** to be adaptable for many languages and cultures

**Localization - L10n**

**Adapt** for many languages and cultures





**Prepare your app**

# A single, flexible layout

A layout that works for all locales

Touch the Google I/O 2013 logo above to sign in and begin using the app.

Tippen Sie auf das Google I/O 2013-Logo, um sich anzumelden und mit der App anzufangen.

上記の Google I/O 2013 のロゴを触ってサインインして、アプリを使用し始めて下さい。





# What's wrong here?

JAVA

```
String getAndroidSales(int numAndroids, String date) {  
    return String.format("%d Android dolls were sold on %s",  
        numAndroids, date);  
}
```



# Using strings.xml

XML

```
<resources>
    <string name="sales">%1$d Android dolls were sold on %2$s
    </string>
</resources>
```

JAVA

```
String getAndroidSales(int numAndroids, String date) {
    Resources res = getResources();
    return String.format(res.getString(R.string.sales),
        numAndroids, date);
}
```



# Using strings.xml

XML

```
<resources>
  <string name="sales">%1$d Android dolls were sold on
    %2$s</string>
  <string name="price">Each doll is sold at %1$d</string>
  <string name="account">Account</string>
  <string name="submit">Submit</string>
  <string name="cancel">Cancel</string>
</resources>
```



# Provide context

XML

```
<resources>
  <!-- The number of Android dolls sold on a particular date.
  "Android" is a trademark, and shouldn't be translated. Try to
  limit to 40 chars.
  -->
  <string name="sales">%1$d Android dolls were sold on
    %2$s<string>
</resources>
```



# XLIFF Placeholders

XML

```
<resources xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2">
  <string name="sales">
    <xliff:g id="number" example="2">%1$d</xliff:g>
    Androids were sold on
    <xliff:g id="date">%2$s</xliff:g>
  </string>
</resources>
```



# Plurals are different across languages

## **One form:** Chinese

- 娃娃

## **Two forms:** English

- 1: doll
- Everything else: dolls

## **Three forms:** Polish

- 1: lalka
- Ends in 2-4, excluding 12-14: lalki
- Everything else: lalek



# Plurality in Breton

- Ends in 1, excluding 1, 11, 71, 91
- Ends in 2, excluding 12, 72, 92
- Ends in 3, 4 or 9, excluding 13, 14, 19, 73, 74, 79, 93, 94, 99
- Ends in 1000000
- Everything else



# Quantity Strings

```
<resources xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2">
  <plurals name="sales">
    <item quantity="one">
      <xliff:g id="number">%1$d</xliff:g> Android was sold on
      <xliff:g id="date">%2$s</xliff:g>
    </item>
    <item quantity="other">
      <xliff:g id="number">%1$d</xliff:g> Androids were sold
      on <xliff:g id="date">%2$s</xliff:g>
    </item>
  </plurals>
</resources>
```





# Quantity Strings

```
<resources>
  <plurals name="sales">
    <item quantity="zero">...</item>
    <item quantity="one">...</item>
    <item quantity="two">...</item>
    <item quantity="few">...</item>
    <item quantity="more">...</item>
    <item quantity="other">...</item>
  </plurals>
</resources>
```



# Quantity Strings

JAVA

```
String getAndroidSales(int numWidgets, String date) {  
    return res.getQuantityString(R.plurals.sales, numWidgets,  
        numWidgets, date);  
}
```



# Android APIs

JAVA

```
import java.text.SimpleDateFormat;

String getAndroidSales(int numAndroids, Date date) {
    Locale userLocale = Locale.getDefault();
    SimpleDateFormat format = getBestDateTimeFormat(userLocale);
    String localizedDate = String.format(userLocale, format,
        date);
    return getString(R.string.sales, numAndroids,
        localizedDate);
}
```



# Android APIs

**Dates:** `java.text.DateFormat`

**Phone Numbers:** `android.telephony.PhoneNumberUtils`

**Decimals, Currencies, Percentages:**

`java.text.DecimalFormat`, `java.text.NumberFormat`



**What movie is this?**



# Breakfast at Tiffany's

When: Thu, May 16, 11am – 12pm ✕

What:   
e.g., Breakfast at Tiffany's

Calendar:

[Edit event »](#)

11 – 12p



# ...translated in Hindi

कब: बृह., 16 मई, 11:00 – 12:00 ✕

क्या:   
उदा., टिफ़नीज़ में नाश्ता

कैलेंडर:

[ईवेंट संपादित करें »](#)

11:00 – 12:00



# Peking Duck

时间: 5月 16日 (周四), 上午10点 – 上午11点 ✕

内容:   
例如: 去全聚德吃烤鸭

日历:

[编辑活动 »](#)

10点 – 11点





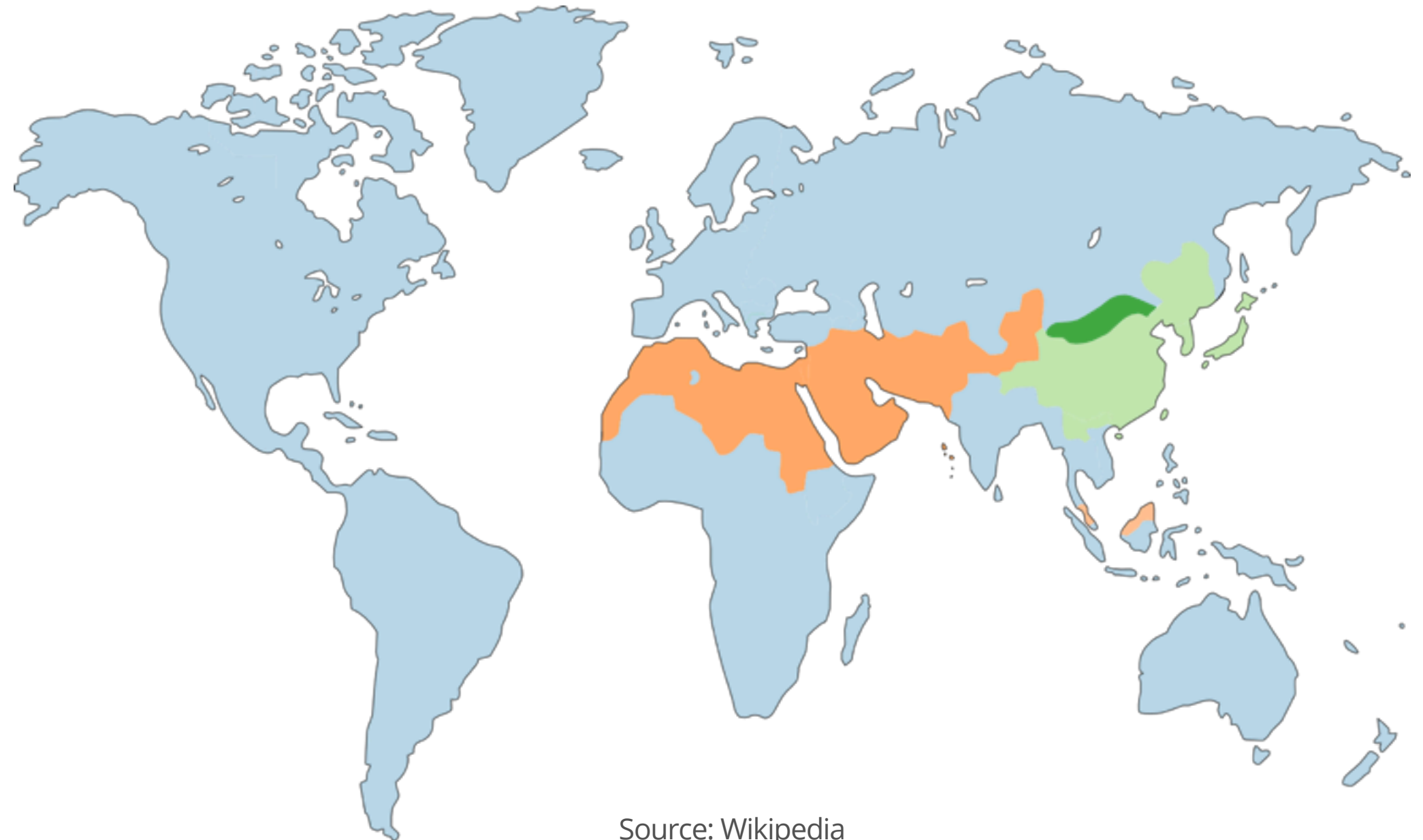
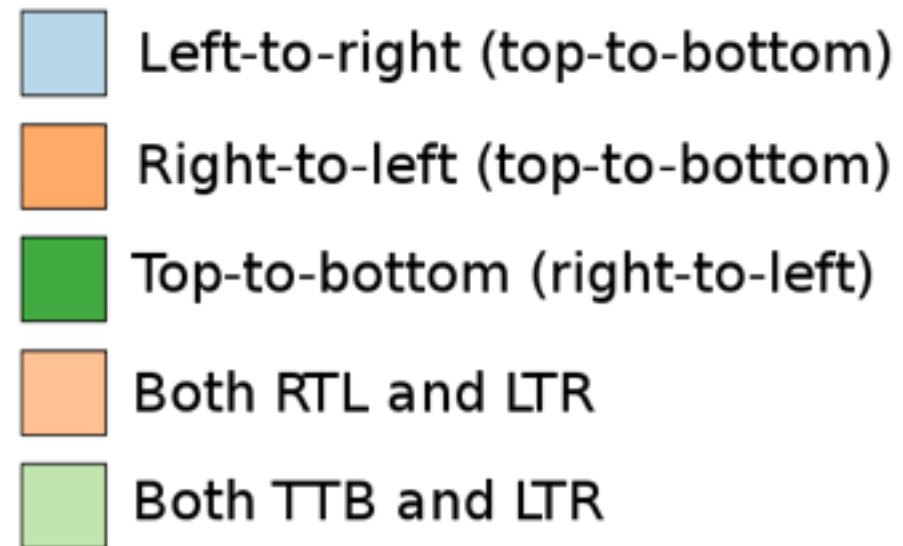


**Advanced: right to left support**

# RTL = right to left

Natural writing direction for Arabic, Persian, and Hebrew languages

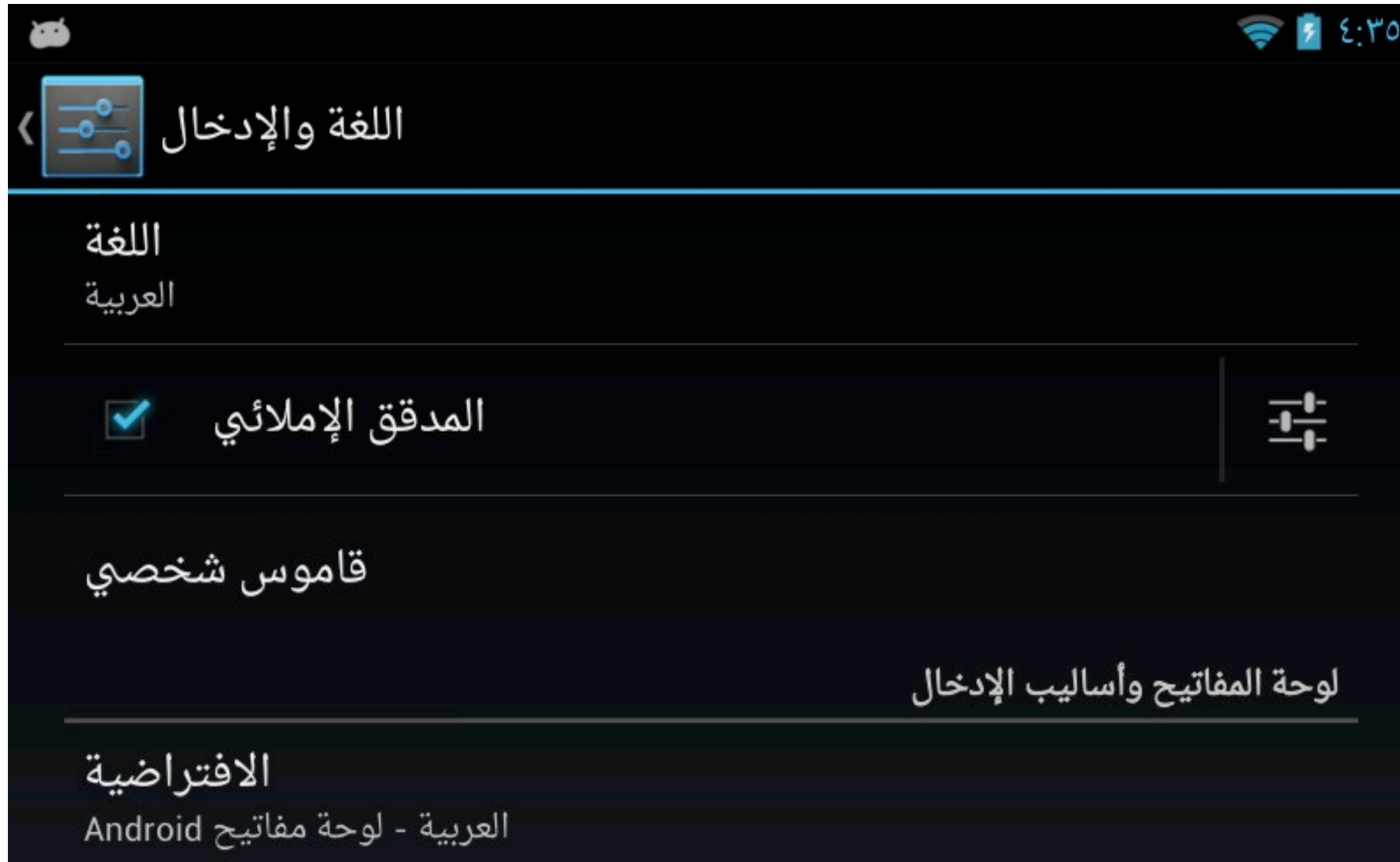
- **300M to 500M** people
- **bug #5597**  
Arabic language support



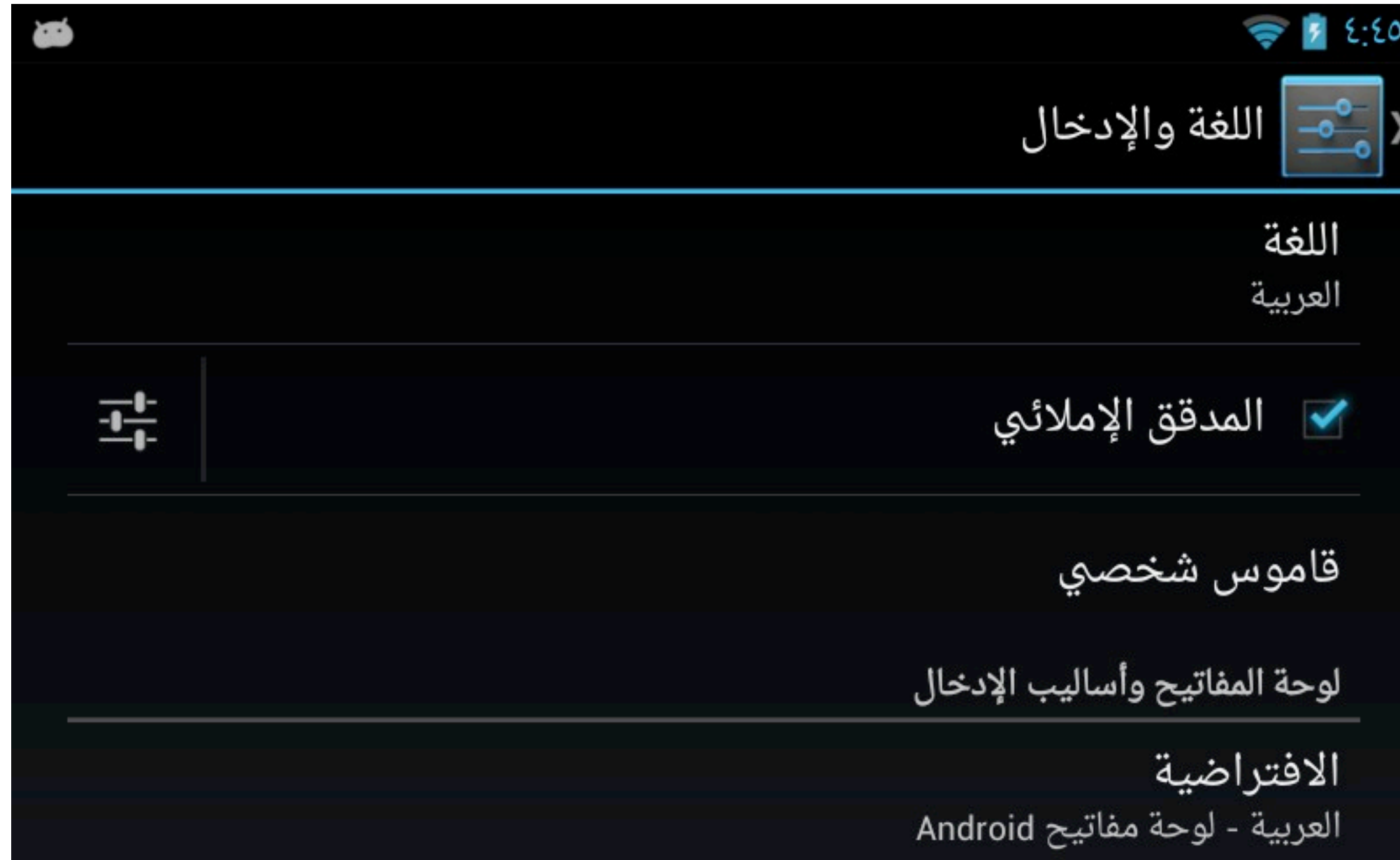
Source: Wikipedia



# What's wrong here?



# What we want: layout mirroring / text alignment



# How?

RTL support built into View and ViewGroup (all widgets too)!

1. AndroidManifest.xml

2. update layouts / resources

3. tune layouts and text

Optional

4. update code for custom views

5. build and test on pre and post JB-MR1

6. validate with native speakers



# Step 1: AndroidManifest.xml

XML

```
<application android:name="MyApp"  
    android:label="@string/myapp_label"  
    android:icon="@drawable/myapp_icon"  
    android:supportsRtl="true">
```

```
and <uses-sdk ... android:targetSdkVersion="17" />
```

```
or <uses-sdk android:minSdkVersion="17" ... />
```



# Step 2: Update layouts

Use start and end properties

- `Gravity.START / Gravity.END`
- `paddingStart / paddingEnd`
- `layout_marginStart / layout_marginEnd`

Resolution process

- API level 17+ : `start / end` override any `left / right`
- API level  $\leq 16$  : `start / end` ignored, `left / right` used



# Resolve start and end - details

| API Level            | 17+                                  | 17+  | 17+                   | ≤16                         | ≤16                                    | ≤16                   |
|----------------------|--------------------------------------|--|-----------------------|-----------------------------|--|-----------------------|
| start / end defined  | Y                                    | Y  | N                     | Y                           | Y                                      | N                     |
| left / right defined | N                                    | Y  | Y                     | N                           | Y                                      | Y                     |
| Result               | start / end resolved to left / right | start / end resolved and override left / right | left / right are used | left = start<br>right = end | left / right used, start / end ignored | left / right are used |





# start / end properties in API level 17

XML

paddingStart  
paddingEnd

layout\_toStartOf  
layout\_toEndOf

drawableStart  
drawableEnd

layout\_alignStart  
layout\_alignEnd

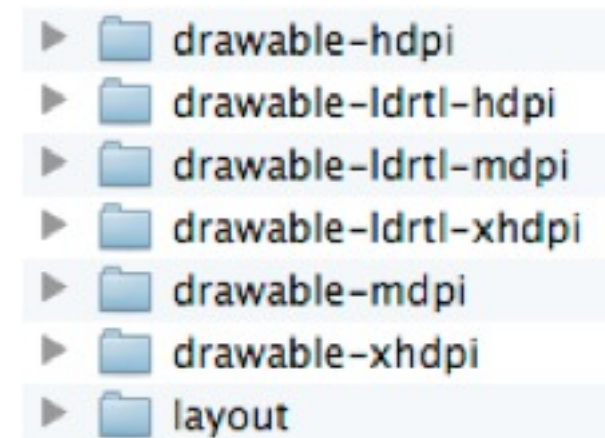
layout\_marginStart  
layout\_marginEnd

layout\_alignParentStart  
layout\_alignParentEnd

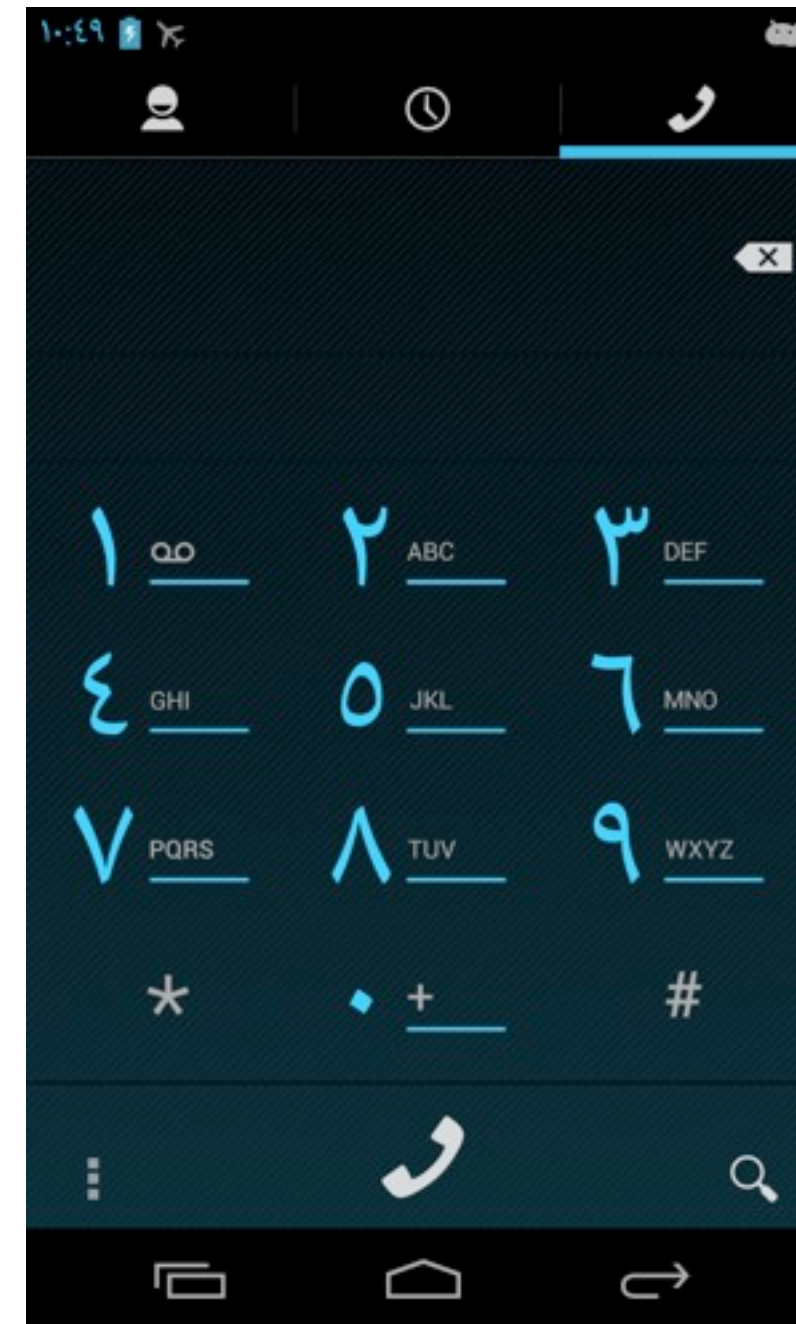
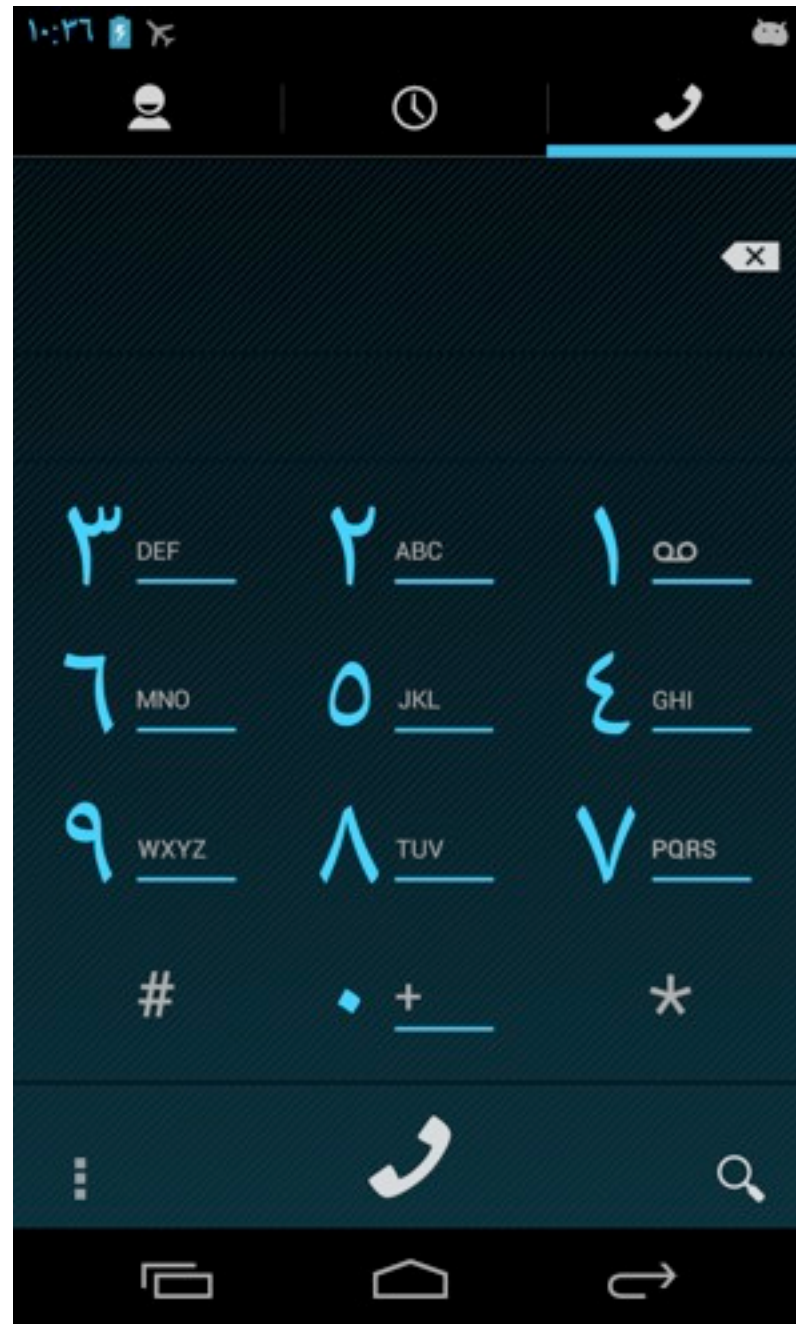


## Step 2: ... Add specific RTL resources

- aapt supports "-ldrtl" resources
- "ld" means layout direction



# Step 3: Tune layouts and text



# layoutDirection

XML

```
<LinearLayout android:orientation="vertical"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:layoutDirection="ltr">
```

## API

```
View.setLayoutDirection(int)  
int View.getLayoutDirection()
```

## Resolution

During `View.measure(...)` and before `View.onMeasure(...)`



# New RTL properties for View

- `layoutDirection`

default : Locale layout direction / LTR for API level  $\leq 16$

- `textDirection`

default : `FIRST_STRONG`

- `textAlignment`

default : `GRAVITY`



# Step 4: update code for custom views

JAVA

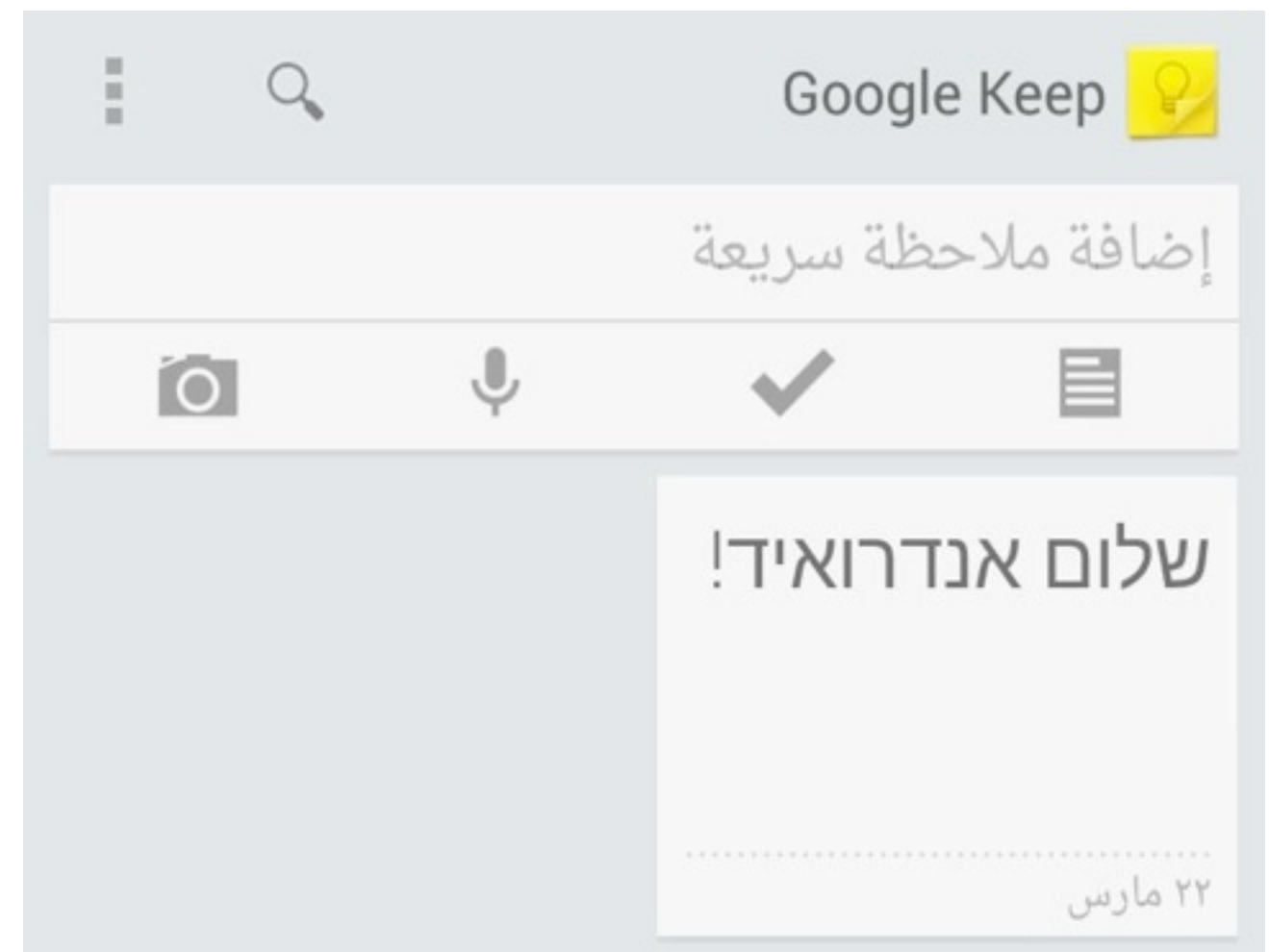
- `Gravity.START / Gravity.END`
- `Gravity.getAbsoluteGravity(int gravity, int layoutDirection)`
- `onRtlPropertiesChanged(int layoutDirection)`
- `onMeasure(wSpec, hSpec)`
- `onLayout(changed, l, r, t, b)`
- `draw(canvas)`



# Compatibility for pre JB-MR1

- all new RTL attributes are ignored
- add start / end properties in addition to the left / right ones
- consider using styles

Great example: Google Keep app



# Tools support for RTL

- aapt with “-ldrtl” resources
- HierarchyViewer

| ▼Text                 |                  |
|-----------------------|------------------|
| getRawTextAlignment() | VIEW_START       |
| getRawTextDirection() | INHERIT          |
| getSelectionEnd()     | -1               |
| getSelectionStart()   | -1               |
| getTextAlignment()    | VIEW_START       |
| getTextDirection()    | FIRST_STRONG     |
| getTextSize()         | 36.0             |
| mText                 | On-site check-in |

| ▼Layout                 |                        |
|-------------------------|------------------------|
| getBaseline()           | 54                     |
| getHeight()             | 64                     |
| getLayoutDirection()    | RESOLVED_DIRECTION_RTL |
| getRawLayoutDirection() | INHERIT                |
| getWidth()              | 256                    |
| hasTransientState()     | false                  |
| isLayoutRtl()           | true                   |
| layout_bottomMargin     | 0                      |
| layout_endMargin        | -2147483648            |
| layout_gravity          | NONE                   |
| layout_height           | WRAP_CONTENT           |
| layout_leftMargin       | 16                     |
| layout_mMarginFlags     | 1                      |
| layout_rightMargin      | 16                     |
| layout_startMargin      | -2147483648            |
| layout_topMargin        | 0                      |
| layout_weight           | 0.0                    |
| layout_width            | WRAP_CONTENT           |
| mBottom                 | 64                     |
| mLeft                   | 352                    |
| mRight                  | 608                    |
| mTop                    | 0                      |

- lint RTL rules in Eclipse plugin





# BidiFormatter for post JB-MR1

- Use case: wrapping string containing user generated content
- Framework and Support Library

JAVA

```
BidiFormatter.getInstance(true /*rtlContext*/).unicodeWrap(phone);
```

|                        |                          |
|------------------------|--------------------------|
| Without BidiFormatter: | הוא עסוק 0000 253 650 1+ |
| With BidiFormatter:    | הוא עסוק +1 650 253 0000 |



# RTL tips and tricks

1. `String.valueOf(int) ==> String.format("%d", int)`

JAVA

2. `getLayoutDirection() == LAYOUT_DIRECTION_RTL`

JAVA

3. GridLayout is better than RelativeLayout

4. `TextUtils.getLayoutDirectionFromLocale(...)`

JAVA

5. SQLite: formatting queries

`String format(Locale.US, String format, Object... args)`

JAVA

6. SQLite: beware of numbers in table / column names








**Translate your app**

# New in Developer Console

## APK TRANSLATION

[Cancel](#)

Choose a service provider that meets your translation requirements.

| SERVICE PROVIDER  | EXPECTED DELIVERY<br>(from receipt of payment) | COST PER WORD | TOTAL    |
|---|--|---------------|----------|
| <br><a href="#">Learn More</a> | 21 May<br>(1 Week)                             | \$0.09        | \$168.57 |
| <br><a href="#">Learn More</a> | 21 May<br>(1 Week)                             | \$0.10        | \$185.43 |
|                                | 21 May<br>(1 Week)                             | \$0.14        | \$262.22 |





**Test your app**

# Layout Tuning

The image shows an IDE interface with XML code on the left and a multi-language preview on the right. The XML code defines a `ScrollView` containing a `LinearLayout` with a `TextView` and a `View`.

```
<ScrollView xmlns:android="http://android:layout_width="match_parent" android:layout_height="wrap_content" android:orientation="vertical" android:background="@color/white" android:paddingLeft="20dp" android:paddingRight="20dp">
  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:background="@color/white"
    android:paddingLeft="20dp"
    android:paddingRight="20dp">
    <TextView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:text="SETTINGS"
      android:textColor="#999999"
      android:textSize="15sp"
      android:fontFamily="sans-serif"
      android:textStyle="bold"
      android:paddingLeft="5dp"
      android:paddingTop="20dp"/>
    <View
      android:layout_width="fill_parent"
      android:layout_height="3dp"
      android:background="#888888"
      android:paddingTop="10dp"/>
  </LinearLayout>
</ScrollView>
```

The preview shows the app's settings screen in eight different languages: English (Ultimate Stopwatch), German (Stopuhr), French (Chronomètre), Hindi (स्टॉपवाच), Norwegian (Stoppeklokke), Thai (นาฬิกาจับเวลา), Vietnamese (Đồng hồ bấm giây), and Chinese (秒錶). Each screen displays settings like 'Sound for Seconds', 'Hand Animations', and 'Vibrate on Alert'.





<http://news.bbc.co.uk/2/hi/7702913.stm>



# Quality testing

APK

**PRODUCTION**  
Version  
**1**

**BETA TESTING**  
Set up Beta testing for your app

**ALPHA TESTING**  
Set up Alpha testing for your app

**PRODUCTION CONFIGURATION** [Upload new APK](#)

**CURRENT APK** uploaded on **May 14, 2013 8:08:09 AM**

**Supported devices**  
**2833**  
[See list](#)

**Excluded devices**  
**0**  
[Manage excluded devices](#)

| ▼ VERSION | UPLOADED ON  | STATUS        |
|-----------|--------------|---------------|
| 1 (1.0)   | May 14, 2013 | Draft in Prod |

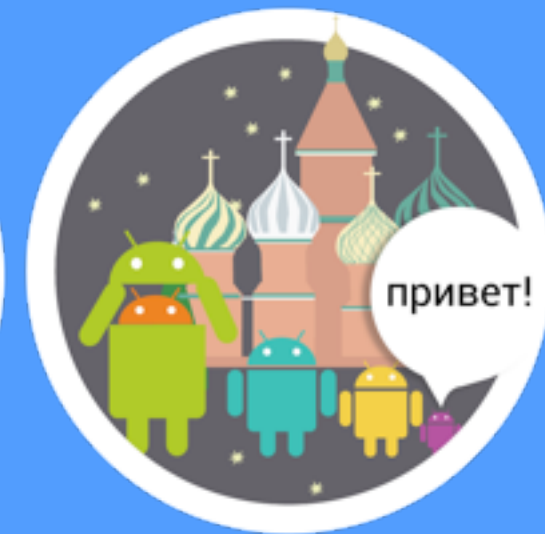




**Last but not least...**

Upload your new APK to distribute in Google Play!





# <Thank You!>



Questions? Come to office hours, or contact us.

Yiwen Zhan [on Google+](#)

Fabrice Di Meglio [on Google+](#)

<https://developers.google.com/international/translation-tools/android>