Mobile I18n Testing Toolbox 2015 Updates

Kat Momoi

Internationalization Engineering Team
Google

International Unicode Conference 39
October 28, 2015

Santa Clara, CA, USA

Supported Locales per OS

- Android Marshmallow: 71 + 2 pseudolocales
    - Lollipop: 71 (+2)
    - KitKat: 56
- iOS 9.1: 43 (w/German; 51 including region variants)
    - iOS 8: 40 (allows arbitrary lang-region locales)
    - iOS 7: 36
- Windows Phone 8.1: 54
    - Windows Phone 8: 50

Cf. Chrome OS: 133

- Many Google web apps support 60-70 locales -- likely to increase further

Challenges for I18n Testing

- How do we test so many locales with a good coverage?
- Not enough test engineers to test all the locales
- Many don’t know how/what to test for i18n

Our Answer …

Mobile I18n Testing Toolbox

- A set of Mobile i18n tools and best practices
- Teams should provide “basic i18n test coverage”

Google

Major items of “Basic” i18n testing

- Functional breakages due to a language specific cause
- Incorrect localized formats:
  - e.g. *date, time, time zones, currency, name format, plurals, gender, sorting order (e.g. contacts list), segmentation, etc.*
- UI breakage due to translation (e.g. text swelling)
- Localizability problems:
  - Hard-coded strings
  - String concatenation
  - Lack of Plurals/Gender
  - … other message formatting issues for translation
  - etc.
- Input issues
  - Compatibility between keyboards and input fields (e.g. CJK IMEs)
  - Data loss

What I18n Testing includes:

but not easy to cover with standard i18n tools or automated tests:

- Special product features
- Natural product workflow in a given locale
    - e.g. Good user experience

What I18n Testing Does Not Cover:

- Translation quality
    - mistranslation
    - less than optimal translation
    - etc.

** Translation quality is best covered by linguists and product evaluators and other evaluation systems

Mobile I18n Testing Toolbox (for Android)

- Tools and Best Practices for mobile i18n testing
  - Multi-locale tests
  - Intl Sanity Checker for locale formats validation (API)
  - Android standard pseudolocales for UI breakages and localizability issues -- “L” and later (Developer Mode)
  - I18n Lint Modules
    - e.g. Layout Testing Tools for auto-detection of UI breakages
  - Keyboard/IME testing tool (API for test + standalone app)
- Where possible (e.g. Android), open source our tools/APIs
  - Open source status toward the end

Multi-Locale Test

What this is good for:

- Check if a different UI language breaks a functionality
  - A typical use case:
    - Switch device locale and check the UI language
  - Another use case:
    - Run some or all available tests with different app locale
      - Provides sanity check on functional regressions due to locale setting

Mechanism for switching locales:

- Switch locale at app level
- Switch device locale
  - Set up multiple devices with different locales

Intl Sanity Checker

- Unit test API on Java/C++ for locale formats
- Easy-to-use comprehensive sanity check API
  - One assert-like method (*assertI18nSanityCheck*) for all purposes
  - Checks all available locale formats by ICU
- Test custom formats built on top of ICU
  - but no need to test what i18n lib provides
- Avoid golden data in locale format tests
  - Use JSON format placeholders like {date}, {time}, {number}, {sorting}, etc. instead of golden data
  - Golden data: Prone to change (e.g. CLDR updates)

Intl Sanity Checker: *Continued*

Usage 1:

- Rather than golden data, use a simple placeholder.
  - {date}: this allows the test to check all available date related formats in ICU

```
assertI18nSanityCheck("<span style='date'>{date}</span>", testObj.actualData(locale), locale); // e.g. "<span style='date'> March 18, 2014</span>" (en-US)

assertI18nSanityCheck("{number}", testObj.actualData(locale), locale); // e.g. "¥1,235" (ja)

assertI18nSanityCheck("{datetime:{skeleton:'MMMMddHmm'}}", testObj.actualData(), ULocale.GERMAN);
//"14. Oktober 8:58"

assertI18nSanityCheck("{sorting}", testObj.actualData(), ULocale.JAPANESE); //"Tokyo,こんにちは,東京"

assertI18nSanityCheck("{phone}", testObj.actualData(), ULocale.US); //"+64 3 331 1234"

assertI18nSanityCheck("{tokenization}", testObj.actualData(), ULocale.JAPANESE); //"自民党,総裁,選挙"
```

Intl Sanity Checker: *Continued*

Usage 2:

- If the actual value as well as the format of data are important for a test
  - E.g. “Date of order: DATE” and “DATE” == “March 18, 2014” for a given locale.
    - Create “Date of order: {date:{value:...}}” pattern in your test. Where the value is a timestamp.

```
assertI18nSanityCheck("Today is {date:{value:1395100800}}", testObj.actualData(), ULocale.US); //"Today
    is March 18, 2014"
assertI18nSanityCheck("{number:{value:1235}}", testObj.actualData(), ULocale.JAPAN); //"¥1,235"
assertI18nSanityCheck("{number:{value:'2013'}}", testObj.actualData(), ULocale.US); //"MMXIII"
```

Intl Sanity Checker: *Continued*

Usage 3:

- Ignore some part of a formatted string
- For example, take a string such as:
  - "Your next card is the Queen of Spades"
    - where the phrase "the Queen of Spades" is returned by a third party random card picker. **Normally a mock is set up for this value.**
  - Instead of a mock, you can use the following pattern "Your next card is {ignore}"

```
assertI18nSanityCheck("Your native name is {ignore}", testObj.actualData(), ULocale.US); //"Your native name is Andris Bērziņš"

assertI18nSanityCheck("Random quote of the day: "{ignore}"", testObj.actualData(), ULocale.UK); //"Random quote of the day: "A horse, a horse, my Kingdom for a horse.""
```

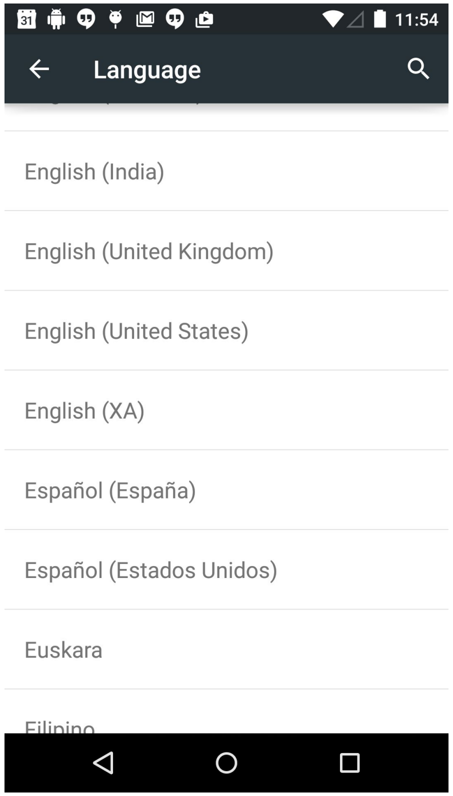
Life of an Android text message

- Create a text message and put it in “strings.xml”
    - ...res/values/strings.xml
- (Place messages into an locale data resource file)
- Send out for translation
- Translations are returned →
    - res/values-fr/strings.xml
- Integrate translations into an application

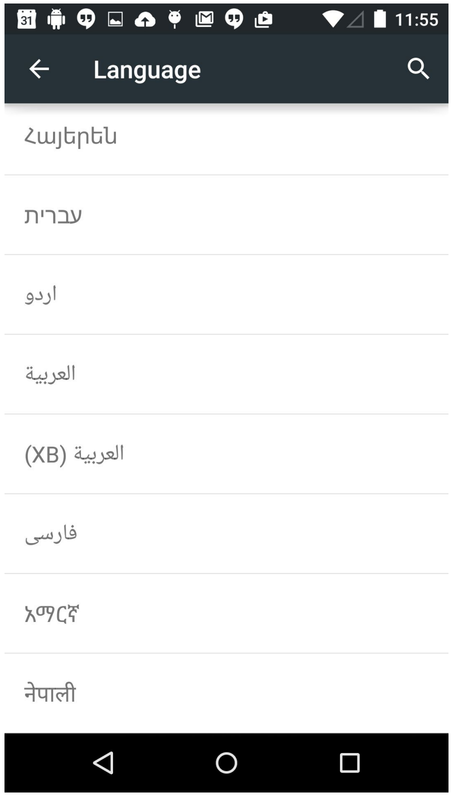
Pseudolocales (en-XA, ar-XB in Android "L")

- Pseudolocales are like any other locale, e.g. fr, de, ja, etc.
- For Android, pseudolocale translations are created from English strings.xml files programmatically
  - Available immediately, no need to wait for (human) translations to come back to test with it
- Pseudolocalizer is part of AAPT build tools [Build Tools Level 21/SDK 23]: new version
- Pseudolocalizer also built into Android Studio [part of SDK release]

Android “L” Language Menu



English (India)
English (United Kingdom)
English (United States)
English (XA)
Español (España)
Español (Estados Unidos)
Euskara
Filipino



Հայերեն
עברית
اردو
العربية
العربية (XB)
فارسی
አማርኛ
नेपाली

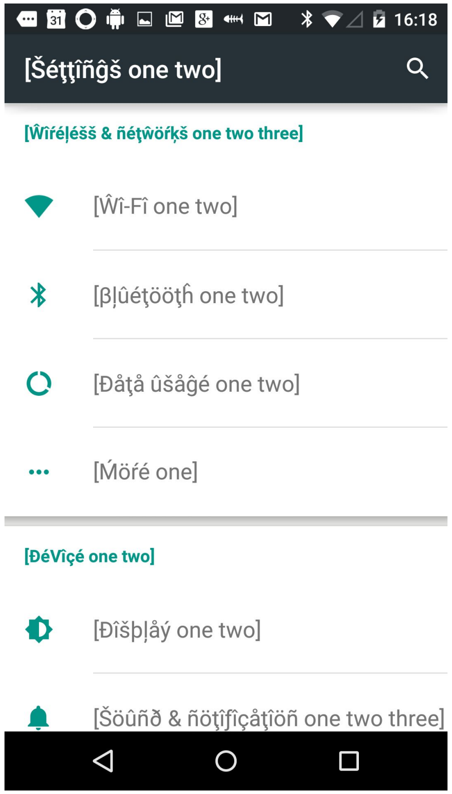
Pseudolocale designs

- LTR Pseudolocale (en-XA) is designed to emulate languages with longer length in translation, e.g. German, Russian, Indic languages, etc.
- RTL Pseudolocale (ar-XB) is designed to emulate RTL languages like Arabic and Hebrew
- Pseudolocales are easy to use if you know English
  - No other language skill is required

LTR Pseudolocale features: English (XA)

- What it looks like:
  - Share what's new… → [Šĥåŕé ŵĥåţ'š ñéŵ... one two three]
- Algorithmically expanded text: (UI Breakage)
  - UI is broken in this locale → quite likely to break in some real locales
- Accents on original text (Localizability)
  - Lack of accents → hard coded strings!
  - No translation when a message is not extracted from code
- The brackets show the boundary of a message. (Localizability)
  - A single sentence is split into 2 or more brackets == an instance of string concatenation: [Ţĥîš îš] [å bööķ]
    - **impossible to translate!**

LTR (en-XA)



16:18

[Šéţţîñĝš one two]

[Ŵîŕéļéšš & ñéţŵöŕķš one two three]

[Ŵî-Fî one two]

[βļûéţööţĥ one two]

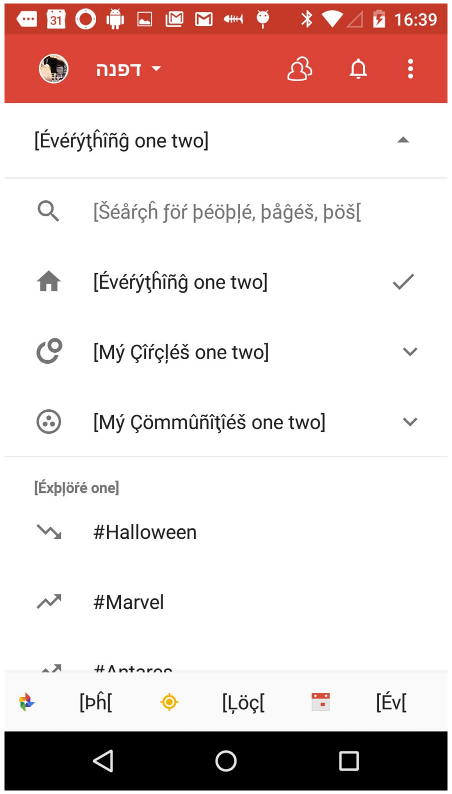
[Ðåţå ûšåĝé one two]

[Ḿöŕé one]

[ÐéVîçé one two]

[Ðîšþļåý one two]

[Šöûñð & ñöţîƒîçåţîöñ one two three]



16:39

דפנה

[Éνéŕýţĥîñĝ one two]

[Šéåŕçĥ ƒöŕ þéöþļé, þåĝéš, þöš[

[Éνéŕýţĥîñĝ one two]

[Mý Çîŕçļéš one two]

[Mý Çömmûñîţîéš one two]

[Éxþļöŕé one]

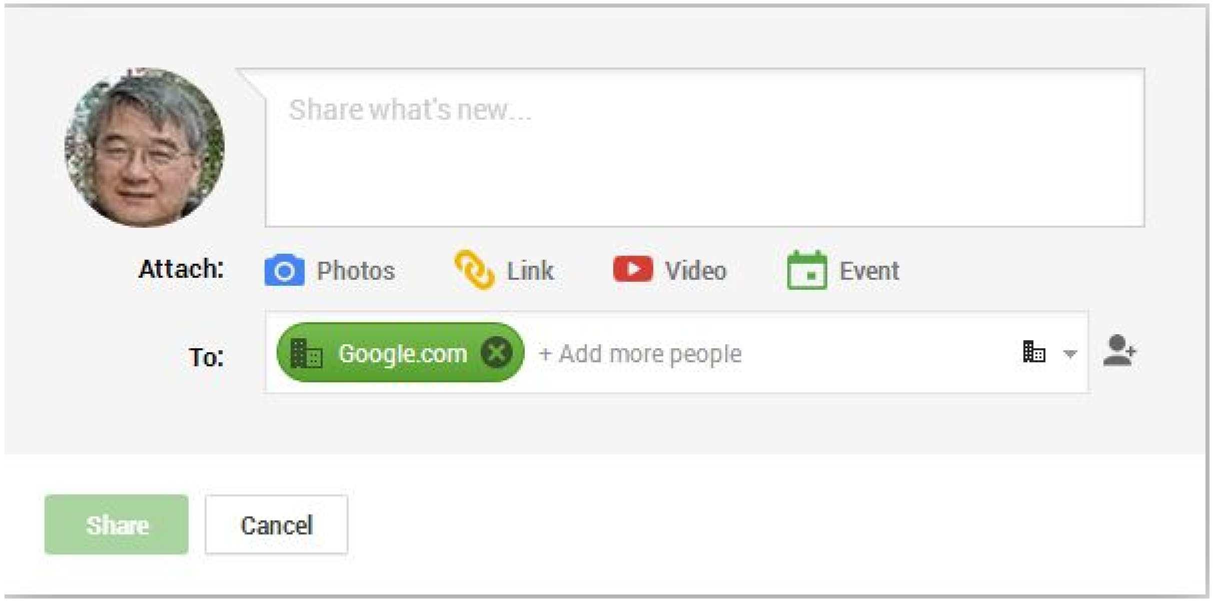
#Halloween

#Marvel

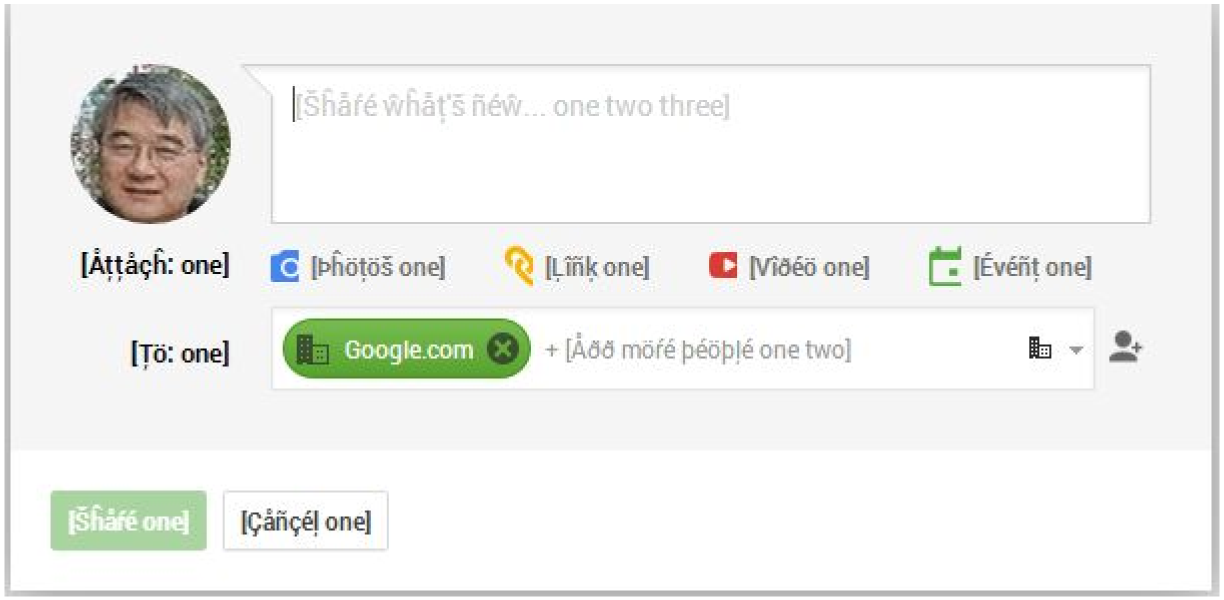
#Antares

[Þĥ[[Ļöç[[Év[

LTR Pseudolocale: Before (en)



LTR Pseudolocale: After (en-XA)



[Šĥåŕé ŵĥåţ'š ñéŵ… one two three]

[Åţţåçĥ: one] [Þĥöţöš one] [Ļîñķ one] [Vîðéö one] [Évéñţ one]

[Ţö: one] Google.com + [Åðð möŕé þéöþļé one two]

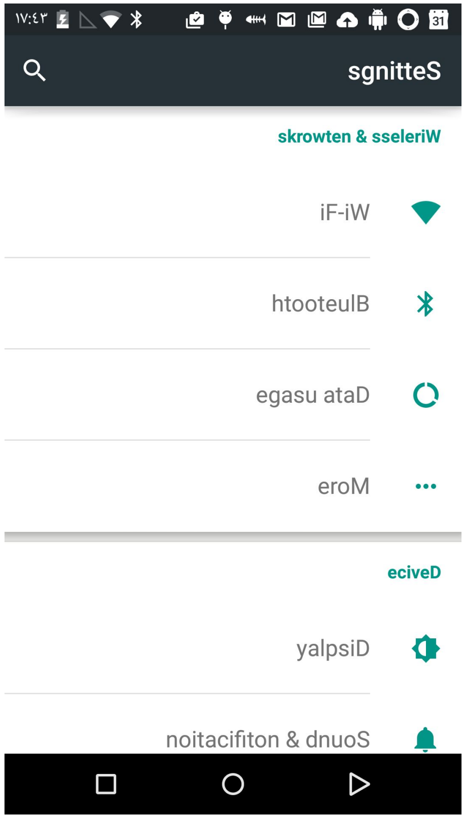
[Šĥåŕé one] [Çåñçéļ one]

RTL Pseudolocale (Fake Bidi) Features

- (XB) العربية
- Built on en locale messages
- Latin Alphabet characters acquire RTL property
    - Text flows right to left
    - "!dlrow olleH" vs. "dlrow olleH!"
- Page elements/objects are mirrored
- OS treats this pseudolocale as an RTL language

Easy to detect anomalies to the above patterns

RTL (ar-XB)


١٧:٤٣

sgnitteS

skrowten & sseleriW

iF-iW

htooteulB

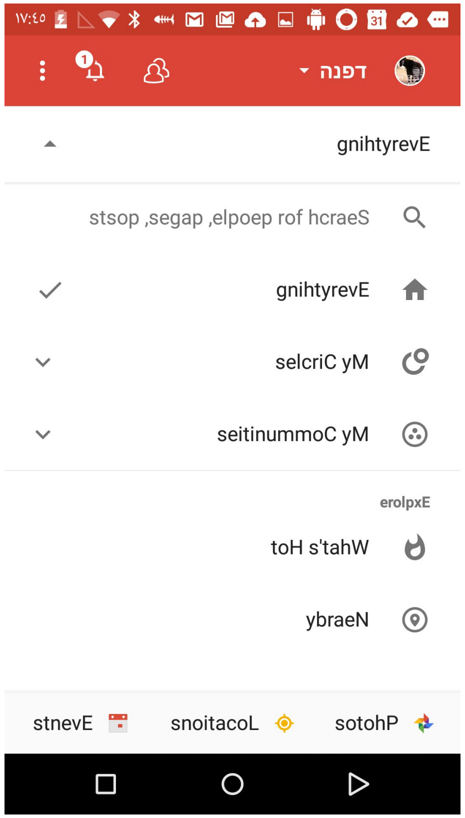
egasu ataD

eroM

eciveD

yalpsiD

noitacifiton & dnuoS


١٧:٤٥

דפנה

gnihtyrevE

stsop ,segap ,elpoep rof hcraeS

gnihtyrevE

selcriC yM

seitinummoC yM

erolpxE

toH s'tahW

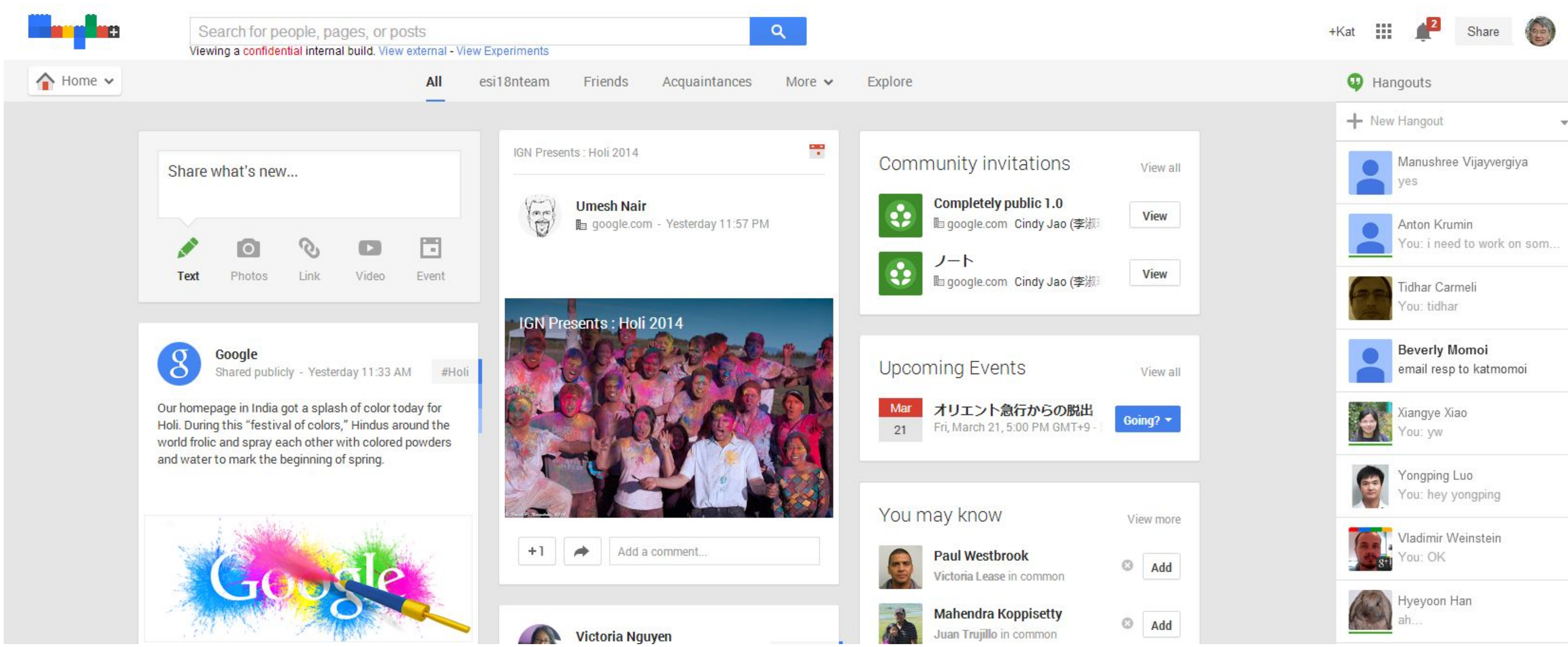
ybraeN

stnevE

snoitacoL

sotohP

RTL Pseudolocale (ar-XB): Before (en locale)



Search for people, pages, or posts

Viewing a confidential internal build. View external - View Experiments

+Kat 2 Share

Home | All | esi18nteam | Friends | Acquaintances | More | Explore | Hangouts

Share what's new...

Text Photos Link Video Event

Google
Shared publicly - Yesterday 11:33 AM #Holi

Our homepage in India got a splash of color today for Holi. During this "festival of colors," Hindus around the world frolic and spray each other with colored powders and water to mark the beginning of spring.

IGN Presents : Holi 2014

Umesh Nair
google.com - Yesterday 11:57 PM

IGN Presents : Holi 2014

+1 | Add a comment...

Victoria Nguyen

Community invitations — View all

Completely public 1.0 — google.com Cindy Jao (李淑 — View

ノート — google.com Cindy Jao (李淑 — View

Upcoming Events — View all

Mar 21 — オリエント急行からの脱出 — Fri, March 21, 5:00 PM GMT+9 - — Going?

You may know — View more

Paul Westbrook — Victoria Lease in common — Add

Mahendra Koppisetty — Juan Trujillo in common — Add

New Hangout

Manushree Vijayvergiya — yes

Anton Krumin — You: i need to work on som...

Tidhar Carmeli — You: tidhar

Beverly Momoi — email resp to katmomoi

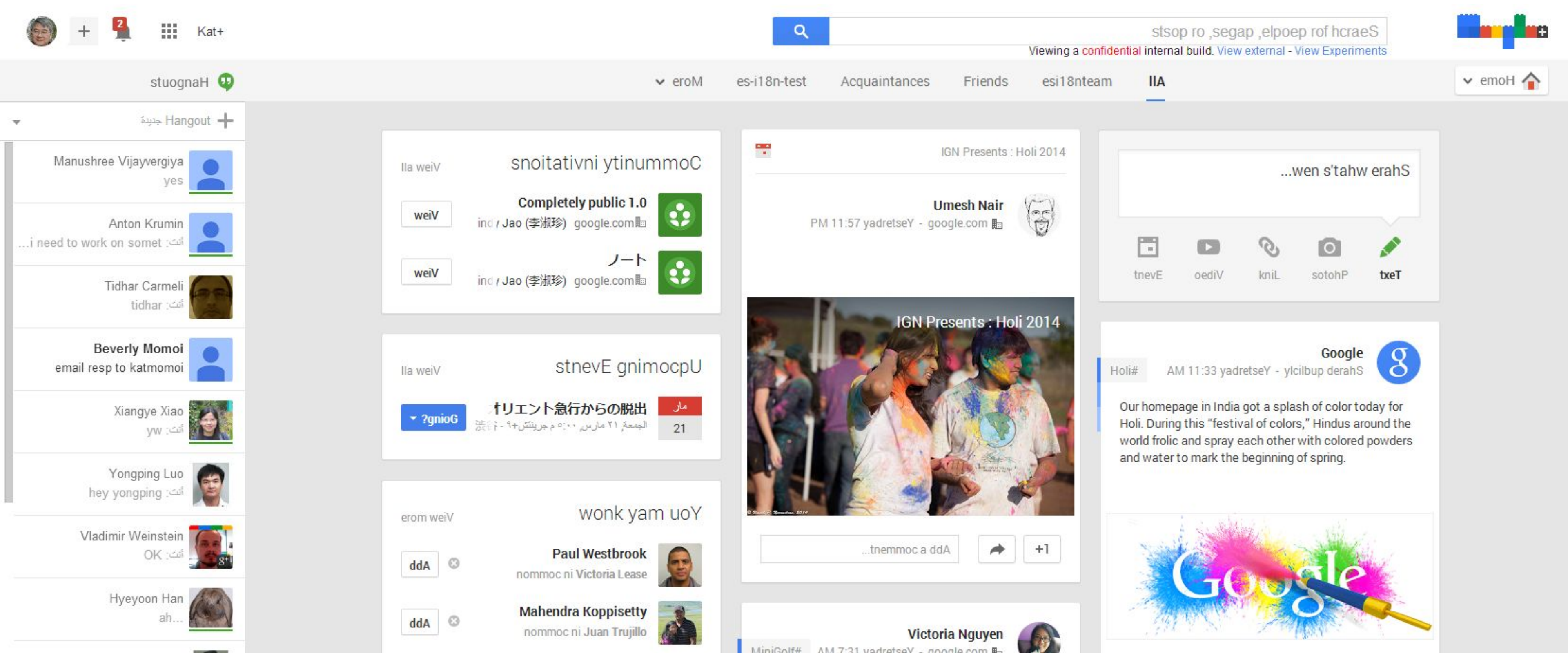
Xiangye Xiao — You: yw

Yongping Luo — You: hey yongping

Vladimir Weinstein — You: OK

Hyeyoon Han — ah...

RTL Pseudolocale: After (ar-XB)



Kat+

stsop ro ,segap ,elpoep rof hcraeS

Viewing a confidential internal build. View external - View Experiments

stuognaH

eroM | es-i18n-test | Acquaintances | Friends | esi18nteam | llA

emoH

جديدة Hangout

Manushree Vijayvergiya
yes

Anton Krumin
...i need to work on somet :أنت

Tidhar Carmeli
tidhar :أنت

Beverly Momoi
email resp to katmomoi

Xiangye Xiao
yw :أنت

Yongping Luo
hey yongping :أنت

Vladimir Weinstein
OK :أنت

Hyeyoon Han
ah...

snoitativni ytinummoC — lla weiV

Completely public 1.0
ind / Jao (李淑珍) google.com
weiV

ノート
ind / Jao (李淑珍) google.com
weiV

stnevE gnimocpU — lla weiV

ナリエント急行からの脱出
الجمعة ٢١ مارس ٥:٠٠ م جرينتش+٩ - 渋谷
مار 21
?gnioG

wonk yam uoY — erom weiV

Paul Westbrook
nommoc ni Victoria Lease
ddA

Mahendra Koppisetty
nommoc ni Juan Trujillo
ddA

IGN Presents : Holi 2014

Umesh Nair
PM 11:57 yadretseY - google.com

IGN Presents : Holi 2014

...tnemmoc a ddA

+1

Victoria Nguyen
MiniGolf# AM 7:31 yadretseY - google.com

...wen s'tahw erahS

tnevE | oediV | kniL | sotohP | txeT

Google
Holi# AM 11:33 yadretseY - ylcilbup derahS

Our homepage in India got a splash of color today for Holi. During this “festival of colors,” Hindus around the world frolic and spray each other with colored powders and water to mark the beginning of spring.

Using pseudolocales

- Regular sanity check for localizability issues (5-10 min)
- Focused use in early stage of new features
- Regular use recommended
  - As UI stabilizes, use it as a sanity check item
- Primary focus on i18n/localizability issues
  - Reduce language specific testing load
    - Dogfood: encourage company-internal community for their native language
  - Coordinate with linguistic check for translation issues
- Recommend: developers to sanity check new UI

iOS App pseudolocale

- For iOS 7 and earlier, two existing locales to be replaced by en-XA and ar-XB translation data files
- Starting with iOS 8, much larger set of languages are supported
  - Still not supporting arbitrary BCP47 locales like en-XA and ar-XB
  - Use rarely used locales like: en-VI (LTR; en Virgin Islands) and ar-TD (Arabic for Chad)
  - Also -- latest Xcode offers 2 pseudolocales
    - Double Length Pseudolanguage
    - Right to Left Pseudolanguage

What may not be pseudolocalized

- User input text
    - comments, posts, all circle names, profile data (including user name)
- Any text that is supplied by i18n libraries/CLDR (for now)
    - Date/Time, Weekdays names, etc.
        - en-XA → same as ‘en’ (English)
        - ar-XB → same as ‘ar’ (Arabic)
    - (Update) Pseudolocalization of CLDR available Google internally
        - Next: open sourcing
- ASCII numbers
- Text/strings from certain server side messages (usually administrative types)
- Text/strings from a different component service

Google Pseudolocales: Odds and Ends

- Brackets “[ ]” mark the boundaries of a translatable message
  - [Ţĥîš îš] [å böök] <-- this is a case of concatenation & a bug
- »...« (□...□) indicates that the string inside the arrows was supplied by the app
  - [⟫**Marco Nelissen**⟪ ⟫⟪öŕîĝîñåļļý šĥåŕéð⟫⟪: one two three four five six]
- Should punctuation marks be pseudolocalized?
  - generally No
  - But there are some cases of this: U+2026
    - “[þĥöţöš one]” → but truncated to “[þĥ[“
      - “[þĥ[“ ←- “[þĥ[...one]

Experimental Features

- CHARLIMIT
    - We might design messages to include CHARLIMIT values as calculated in ASCII characters by the developer
- We might experiment with the following:
    - If pseudo text expansion is within CHARLIMIT, then use as is
    - If pseudo text expansion is longer than CHARLIMIT, then use CHARLIMIT for pseudo expansion lengthk

Pseudolocale Deployment Notes

Prep internal dev environment for en-XA, ar-XB:

- Standardize on these 2 locale names (on par with any ordinalry locales)
  - BCP 47 compliant
  - Ensure that programming languages can process these 2 locales (BCP 47 compliant) -- remove any blockers
  - Add them to build config
  - Persuade all projects to use only these locale names
- Make it super easy to build apps with these 2 pseudolocales, e.g.
  - Option 1: (Android only) Create pseudolocales with AAPT build option
  - Option 2: Automatically add data files for 2 pseudolocales at every source update
    - Create them out of the master English file
    - Build with these 2 locales added to the build config
  - Option 3: If using Pseudolocalizer, make it a default to produce these 2 locales

IME/Keyboard Test

- Reliable check for compatibility problems between IME/Keyboard and a given input field in an app
- Typical problem: CJK input is prematurely committed making it impossible to enter correct characters
- Typical causes:
  - Listener code is added to an input field in such a way to interfere with an IME
  - Additional feature such as “auto-suggest” is added to an input field and interferes with normal keyboard input, especially for CJK IMEs

Android Standalone Mock IME App

- Android standalone Mock IME app:
  - Test essential keyboard actions against a given input field
    - Checks if your app is compatible with various keyboards including CJK IMEs
  - Takes 30 seconds or so to run the essential compatibility tests on any input field
    - A sanity check test when an input field has been modified
- (Update): Now a Chrome IME app

Android IME Test

- Android IME test: can run on Espresso
  - Uses a mock IME and avoids the instability of real IME/keyboards
    - There are many keyboards!
  - Run it on any input fields with something on top of vanilla input box, e.g. added listener, auto suggestion feature, etc.

Mock IME App demo

- Chrome on Android
  - Movie 1
- Google Keep on Android
  - Movie 2

This app is open sourced in AOSP

Layout Tests

- Added to Android code lint
- Coming in future:
    - a set of automated layout breakage tests that run on Espresso -- on the pseudolocale, en-XA

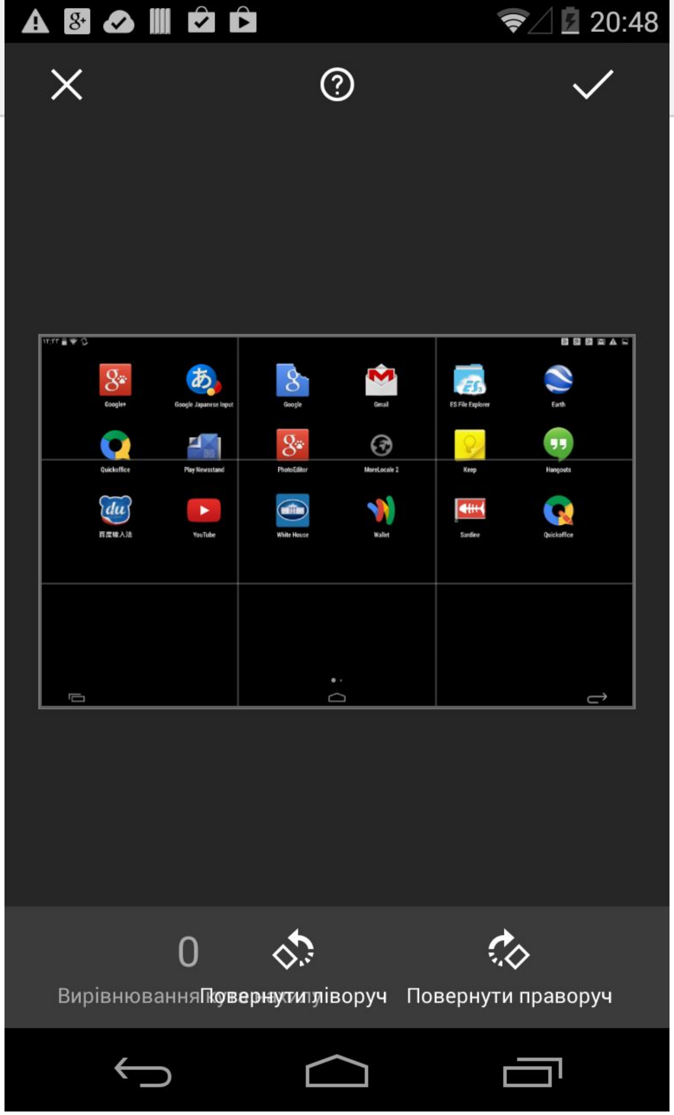
Lint for Layout Breakage check

```
$ lint --check RelativeOverlap

...photoeditor/res/layout/*.xml

Scanning photoeditor: ............
res/layout/filter_parameter_fragment.xml:71: Warning:
@id/third_tool_button can overlap LinearLayout-1 if LinearLayout-
1, @id/third_tool_button grow due to localized text expansion
[RelativeOverlap]
    <...photoeditor.views.ToolButton
    ^
0 errors, 1 warnings
```

Photoeditor in Ukrainian



Summary of Android I18n Lint Modules: 1

- **RTL Compatibility** - checks that your application does not use *left* and *right* constants, i.e. incorrect for RTL languages. Use *start* and *end* instead.
- **Hardcoded text in layout files** - reference translatable strings in strings.xml instead of hard coded messages for your UI, e.g. text=”@string/helloworld” with an entry in strings.xml
- **Message correctness** - check for placeholders match in all translations and quantity variants match translation language. E.g. check that Russian translation of a plural has four variants (one, few, many, other)
- **RelativeOverlap** --- check for object overlaps

Summary of Android I18n Lint Modules: 2

- **Hardcoded text in setText** (new) - checks that application does not pass a hardcoded string into a UI widget setText method.
- **String concatenation in setText** (new) - checks that string concatenation is not used when calling UI weiget setText method. Concatenated strings are not translatable in most cases.
- **Number formatting in setText** (new) - checks that methods like Integer.toString() or Long.toString() are not used for number formatting when displaying text to user. These methods format numbers using ASCII digits only, while locale may require native digits (e.g. Arabic). Use String.format instead.

Open Sourcing (updated: 10/2015): 1

- New Pseudolocalizer
  - Android SDK 23 (with L) and above
    - Part of AAPT Build tools: *aapt --pseudo-localize*
  - Also part of latest Android Studio
    - *“File” > “Project Structure” > "Build Types" > "Pseudo Locales Enabled"*
- Android Lint (in next Android SDK release - Oct/Nov 2015?)
  - RTL Compatibility
  - Hardcoded text in layout files
  - Message correctness
  - RelativeLayoutOverlap (starting in SDK 24)

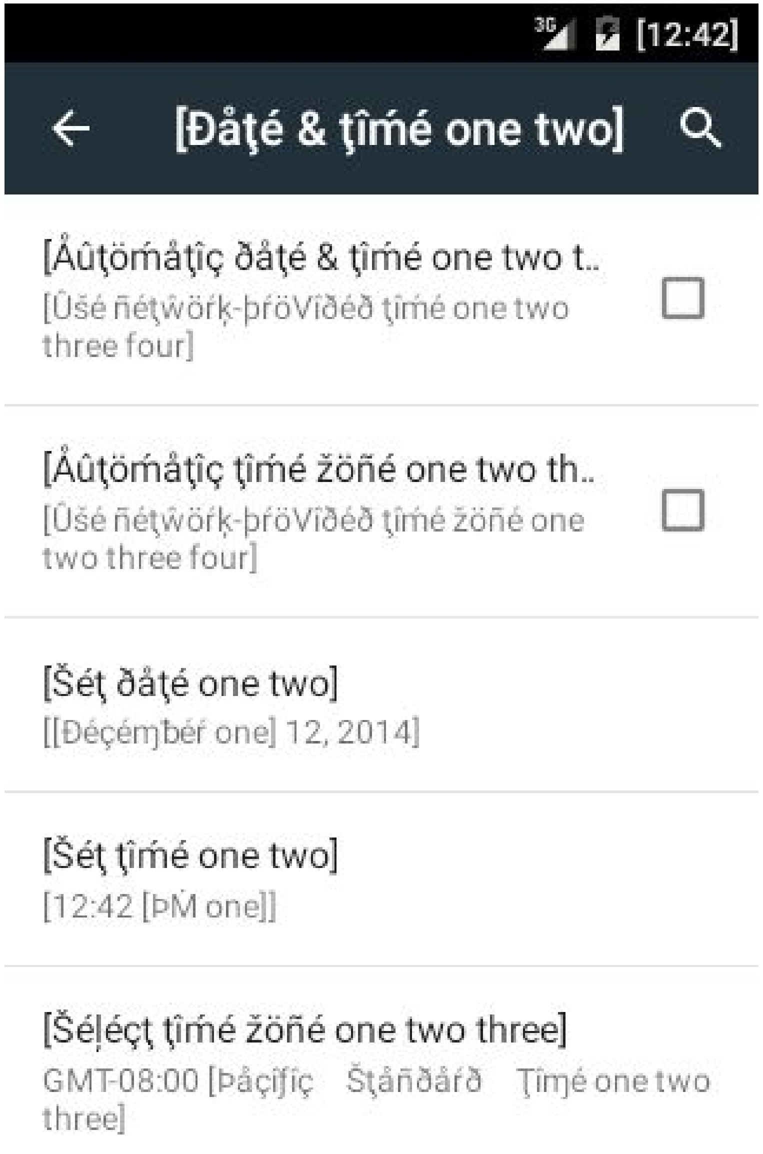
Open Sourcing (updated: 10/2015): 2

- Mock IME test code and standalone app

  In an AOSP package -- pre-installed in emulators & source code (in SDK 24)
  - Mock IME Test also uses Mock IME app
  - Chrome app version (internal only at present) -- planning open sourcing
- Intl Sanity Checker:
  - Github: https://github.com/googlei18n/i18n_sanitycheck
    - Java version (available)
    - C++ (open sourcing planned)
- Pseudolocalization of CLDR: in progress
- Android String Override: development complete. Open source planned
  - Allows user to override string resources in apps
  - Edit translated messages in situ

Google

CLDR Pseudolocale


[12:42]

[Ðåţé & ţîmé one two]

[Åûţömåţîç ðåţé & ţîmé one two t..
[Ûšé ñéţŵörķ-þröVîðéð ţîmé one two three four]

[Åûţömåţîç ţîmé žöñé one two th..
[Ûšé ñéţŵörķ-þröVîðéð ţîmé žöñé one two three four]

[Šéţ ðåţé one two]
[[Ðéçémbéŕ one] 12, 2014]

[Šéţ ţîmé one two]
[12:42 [ÞṀ one]]

[Šéļéçţ ţîmé žöñé one two three]
GMT-08:00 [Þåçîƒîç Šţåñðåŕð Ţîmé one two three]

CLDR Pseudolocalization Specifics

Yes:

1. Languages, territories and variants names
2. Month and weekdays names (various calendars)
3. Datetime formatting patterns (brackets only)
4. Time zones
5. Relative date and time (yesterday, tomorrow, in N days, last Sunday)
6. List patterns ([a, b åñð c one])

No:

0. nothing in root.xml
1. Unit names (mile, fahrenheit)
2. Currencies
3. Calendars

Android String Override Tool: what is it?

Allows overriding of resource strings on a running app

- Does not require any code modifications or special build
- Works with any Android application

Potential uses:
- Adding invisible characters into overriden strings and determine what exact strings are used on a given screen
- Expand/double original strings to check if application UI is flexible enough to support translation to different languages. It allows to find UI limitations or potential issues before translations are available

.

Android String Override: how it works

0. Run the override app
1. Run a setup script passing name of app to override
2. Push a text file with override strings to the device.
3. When the app redraws its content, overriden values are used.
4. To try out a new string value, update the entry in the override file
    - Push it to device. The change is applied automatically

Conclusions

Mobile I18n Testing Toolbox

- Designed to address the major areas of i18n testing
- Time saving and efficient
- More features are planned ...

Q&A

Questions/Comments?

Internationalization (I18n) vs. Localization (L10n)

Internationalization is enabling your product code so that it can be localized/translated into different languages without modifying the core code

- I18n is design

Localization provides translation of messages, charts/graphs, and images for supported languages/locales without touching the core product code

- Localizable resources must be extracted from code